Distinguishing Resistance-equivalent Isomorphic Graphs via Spectral-based Graph Representations

anonymous

***Abstract*—Graph Neural Networks (GNNs) usually face an expressive power bottleneck, limiting their capability to distinguish pairs of graphs that are structurally similar yet non-isomorphic. Very recently, Graphormer-GD, one of state-of-the-art models, leveraged *resistance distance* to represent graph topological features and effectively solved the problem. However, the resistance distance feature fails to distinguish *resistance-equivalent graphs*. To address the fundamental challenge, a new representation method, using spectral features—the eigenvectors of the graph Laplacian—to capture the topological information that resistance distance overlooks, is proposed in this study, showing the discriminability when distinguishing representative resistance-equivalent graphs. To demonstrate the capability of our embedding method on real world datasets, a model, called Spectral-Graphormer, integrating spectral features into the Graphormer architecture, is also proposed. The model outperforms the original Graphormer, and it performs comparably to Graphormer-GD, and surpasses both on the large-scale benchmark, achieving test accuracy comparable to MPNN baselines such as PNA. In particular, rather than constructing a memory-intensive $N \times N$ resistance-distance matrix, the model builds an $N \times k$ spectral-embedding matrix, which significantly reduces memory usage—especially in large-scale graphs. Apparently, spectral features provide a more robust and scalable structural encoding for graph representations. The proposed approach is theoretically grounded and practically effective for distinguishing non-isomorphic graphs with similar structures, and it further demonstrates significant merits in large-scale graph tasks. The datasets and code used in this work are available at https://github.com/LeiaHsieh/Spectral-Graphormer.**

***Index Terms*—Graph Representation, Graph Isomorphism, Spectral Methods, Laplacian eigenvectors, Resistance Distance, Graph Neural Network, GNN**

I. Introduction

Graphs have recently been used as a powerful tool for representing various structured and complex data, including social networks, knowledge graphs, protein–protein interaction networks, and so on. Graphs can model complex relationships between entities; for example, encoding a social network as a graph, where nodes represent individual users and edges represent the relationships between nodes, such as friends or colleagues [1], [2]. As a general form of data organization, graph representation, also known as graph embedding, has emerged as a crucial research area in data science and machine learning. One can exploit a graph representation to convert high-dimensional, sparse structured data into low-dimensional, dense vectors such that the resulting vector representations (i.e., embeddings) capture and preserve the structural properties of the original data. In this study, we focus on embedding simple and connected graphs. A *simple graph* is defined as an undirected graph with no self-loops or multiple edges, while a *connected* graph ensures that there is a path between any two vertices [3].

However, extracting effective representations from a given graph poses a significant challenge. A truly effective representation method must assign distinct embeddings to non-isomorphic graphs. Otherwise, models may fail to distinguish between structurally different graphs. That is, two non-isomorphic graphs may exhibit similar features, leading to ambiguous interpretations and confusion in downstream tasks. Here, two graphs, $G = (V_G, E_G)$ and $H = (V_H, E_H)$, are *isomorphic* if there exists a bijective mapping $f : V_G \to V_H$ such that $(u, v) \in E_G$ if and only if $(f(u), f(v)) \in E_H$. Figure 1a and Figure 1b illustrate examples of isomorphic and non-isomorphic graphs, respectively. This raises a key question of whether there exists an embedding method that guarantees to yield distinct representations for non-isomorphic graphs.

One can observe that many basic graph properties, such as the number of vertices, the number of edges, and degree sequence—are seldom sufficient to distinguish structurally dissimilar graphs. The issue can be illustrated with a toy example, as shown in Figure 2. The two graphs have the same basic properties, but their structures are different; i.e., they are non-isomorphic. The example shows that the graph invariants are insufficient to distinguish non-isomorphic graphs.

Therefore, we aim to develop an embedding strategy that maps structurally distinct graphs to distinct representations. The objective is to ensure that if G_1 and G_2 are non-isomorphic, $G_1 \ncong G_2$, then their embeddings must be unequal: $\mathbf{emb}(G_1) \neq \mathbf{emb}(G_2)$. Note that $\mathbf{emb}(G_1) \neq \mathbf{emb}(G_2)$ usually guarantees that G_1 and G_2 are different, but the converse is challenging. Our work thus focuses on identifying a more discriminative graph spectral approach to address this core challenge in graph representation.

(a) Isomorphic graphs (b) Non-isomorphic graphs

Fig. 1: An illustration of isomorphic and non-isomorphic graphs

Fig. 2: Two non-isomorphic graphs with the same number of vertices (6), edges (6), and degree sequence $(1, 1, 2, 2, 3, 3)$.

Our Contributions. We propose Spectral-Graphormer, a GNN model that enhances graph representations by integrating spectral features into the powerful Graphormer architecture [4]. The key contributions are summarized as follows:

- **Demonstrating the Limits of Resistance Distance:** We present a challenging class of non-isomorphic graphs that are resistance-equivalent; that is, exploiting resistance distances fails to distinguish these graphs, but our spectral approach can successfully achieve the task.
- **Proposing a Scalable Spectral-based Model:** We introduce Spectral-Graphormer, which integrates spectral features (the eigenvectors of a graph Laplacian matrix) into the Graphormer architecture. This approach not only captures the subtle topological information that resistance distances may miss but also significantly improves memory efficiency by replacing a dense $N \times N$ resistance matrix with a lean $N \times k$ spectral embedding. Therefore, our model is particularly well-suited for large-scale graph datasets, e.g., OGBG-PPA, where some previous methods faced memory bottlenecks, like: distance-based models (e.g., Graphormer-GD [5]) and subgraph-based GNNs (e.g., SUN [6]).
- **Achieving Strong Empirical Performance:** We validate our model on public benchmarks. On graph regression (ZINC) and graph classification (OGBG-PPA) tasks, we show that Spectral-Graphormer outperforms the original Graphormer [4] and it achieves competitive or superior performance compared to strong baselines, including Graphormer-GD, PNA [7], and SUN. This confirms that our spectral approach is both theoretically powerful and practically effective.

II. Related Work

Distinguishing between non-isomorphic graphs remains a core issue within graph theory and graph representation learning. A widely adopted theoretical reference for evaluating the expressive capabilities of Graph Neural Networks (GNNs) is the classical Weisfeiler-Lehman (WL) test [8]. However, previous studies [9], [10] presented that the capability of most Message-Passing Neural Networks (MPNNs) is upper-bounded by the 1-WL test. Precisely, any pair of non-isomorphic graphs that the 1-WL cannot distinguish can also be indistinguishable from these GNN architectures. The challenge inspired a considerable amount of studies to design more expressive graph representations.

Several recent breakthrough works centered around the properties of *resistance distance*. In graph representation learning, Zhang et al. [5] explored the expressive capacity of graph representations through the concept of graph biconnectivity, with a particular emphasis on resistance distance. They focused on leveraging the features of resistance distance to enhance the expressiveness (see Section II-A). Meanwhile , from the graph-theoretic perspective, Xu et al. [11] investigated the properties of resistance-equivalent graphs (see Section II-B).

A. Expressive Power of Resistance Distance

The Resistance Distance (RD), a classic metric in graph theory [12], [13], [14], provides a means of capturing the global structure of a graph. Conceptually, the RD between two vertices i and j represents the effective resistance in an equivalent electrical network in which each edge corresponds to a unit resistor. This metric considers all paths between the vertices, making it highly sensitive to structural bottlenecks such as articulation vertices or bridges. The RD value represents how "easily" current flows between vertices through all available paths.

Recent work by Zhang et al. [5] systematically investigated the expressiveness of GNNs. They demonstrated that most existing graph representation frameworks fail to distinguish between certain non-isomorphic graphs, particularly those differentiated by biconnectivity properties, such as the presence of cut vertices (articulation points) or cut edges (bridges)—a core concept in graph theory [15].

To highlight this weakness, Zhang et al. [5] constructed challenging pairs of graphs in which one graph contains cut vertices or edges while the other does not. These pairs were carefully designed to share identical basic invariants—such as vertex and edge counts and degree sequences—and similar local neighborhood patterns, making them indistinguishable from most GNNs. Their findings confirmed that the resistance distance can successfully distinguish the challenging pairs. They further proposed the Generalized Distance Weisfeiler-Lehman (GD-WL) framework [5], which integrates a general distance metric, such as resistance distance, into the iterative color refinement procedure, allowing it to capture the graph's topological features more effectively.

B. Expressive Limits of Resistance Distance

Despite the strengths of RD, its limitations is worth exploring. In fact, the limitation comes from the class of resistance-equivalent graphs: non-isomorphic graphs that have identical resistance spectra, denoted by $\mathrm{RS}(G)$, a multiset of all pairwise resistance distances. Figure 3 shows an example.

A very recent work by Xu et al. [11] presented a method for constructing families of the resistance-equivalent graphs, which are non-isomorphic but possess the same resistance spectrum. Their method utilized the concept of integer partitions with equal sums of squares. Their construction method is based on an S-resistance-transitive graph. That is, given a positive integer t, they select two distinct partitions whose sums of squares are equal (i.e., $\sum_i a_i^2 = \sum_j b_j^2$). They then group the t attachment graphs according to the partitions and apply a combined procedure to produce non-isomorphic graphs with the same resistance spectrum.

Apparently, for resistance-equivalent graphs, an embedding or a graph feature that relies only on the collection of resistance distances produces exactly the same output. The limitation motivates us to seek a better embedding strategy, especially for resistance-equivalent graphs. In particular, our work addresses this gap by exploring spectral features, aiming to capture the topological characteristics of a graph more precisely. We believe the proposed spectral approach could pave the way for distinguishing highly similar non-isomorphic graphs.

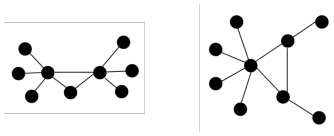
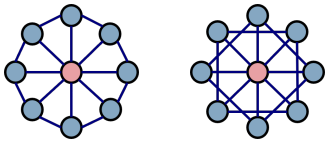


Fig. 3: Two resistance-equivalent graphs with the same resistance spectrum: $\mathrm{RS}(G) = [\frac{2}{3}]^3[1]^6[\frac{5}{3}]^{12}[2]^6[\frac{8}{3}]^9$, where $[x]^n$ denotes n occurrences of x (as shown in Xu et al. [11])



(a) Graph 1 (b) Graph 2

Fig. 4: An illustration by the pair of graphs in the *m1k4* case [5], where the two graphs are obviously non-isomorphic

C. RD-based representation

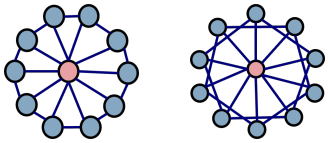
We revisited the topological features of RD used in Graphormer-GD [5]. RD actually models a graph as an electrical network in which each edge represents a unit resistor; the resistance distance between two vertices represents the effective resistance between them. This concept was first formalized in the seminal work of Klein and Randić [13], and later extended with important structural properties and sum rules [16]. Unlike the shortest path distance, resistance distance considers all possible paths, allowing it to capture global topological information more effectively than the shortest path distance. We applied the resistance distance and the shortest path distance to the *m1k4* and *m1k5* cases, discussed in [5] (see Figures 4 and 5). The shortest path distance failed to distinguish between the two non-isomorphic graphs in each case, as the end-to-end shortest path distances (SPD) were identical across the graph pairs. In contrast, the resistance distance successfully differentiated between the graphs in both the *m1k4* and *m1k5* cases, demonstrating its power over centrality measures on these deliberately challenging graphs designed by Zhang et al. [5] Although resistance distance is usually considered more expressive than shortest-path distance, one may ask whether there exist pairs of non-isomorphic graphs that are indistinguishable under the resistance-distance metric.

To answer the question, very recently Xu et al. [11] studied resistance-equivalent graphs (see Figure 6 and Figure 7), which indicate that the resistance spectra of such pairs of graphs are identical.

III. Spectral-based Representation

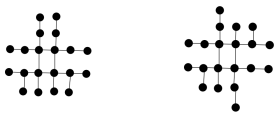
Extracting distinct representations for graphs that appear structurally similar yet are non-isomorphic is a non-trivial task. Our research began with an exploratory question: Can a topology-based graph representation method effectively capture these structural differences?

The existence of resistance-equivalent graphs motivated us to search for a more fundamental and expressive feature than resistance distance. We ultimately found a solution in the foundational theory of graph theory: spectral methods.



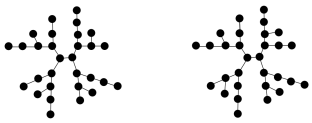
(a) Graph 1 (b) Graph 2

Fig. 5: Another example from the *m1k5* case in [5], where the two graphs are structurally very similar to m1k4, each of which contains two additional vertices, and they are not isomorphic, either



(a) Graph 1 (b) Graph 2

Fig. 6: A pair of non-isomorphic resistance-equivalent graphs in [11] with the same resistance spectrum: $\mathrm{RS}(G) = [\frac{3}{4}]^4[1]^{18}[\frac{7}{4}]^{16}[2]^{22}[\frac{11}{4}]^{32}[3]^{24}[\frac{15}{4}]^{32}[4]^{18}[\frac{19}{4}]^{16}[5]^8$, where $[x]^n$ denotes n occurrences of x



(a) Graph 1 (b) Graph 2

Fig. 7: Another example from [11], indicating that the two non-isomorphic resistance-equivalent graphs have the same resistance spectrum: $\mathrm{RS}(G) = [1]^{29}[2]^{38}[3]^{50}[4]^{64}[5]^{78}[6]^{82}[7]^{64}[8]^{26}[9]^4$, where $[x]^n$ denotes n occurrences of x

In algebraic graph theory, the eigenvalues and eigenvectors of a graph's matrices—such as the adjacency matrix or Laplacian matrix—encode rich structural information [17], [18]. These spectral properties are connected to fundamental graph characteristics such as connectivity, the presence of cycles, community structure, and expansion. For instance, the multiplicity of the eigenvalue zero in the Laplacian corresponds to the number of connected components, while the second smallest eigenvalue (known as the Fiedler value) is closely related to the graph's conductance and cut properties [19], [20]. These relationships form the basis of spectral graph theory [21].

We propose constructing a graph's representation by calculating the eigenvalues of its adjacency matrix and its normalized Laplacian matrix:

- **Adjacency Spectrum**: Global connectivity patterns and local features, such as cycles, are reflected in the eigenvalues of the graph's adjacency matrix $\mathbf{A}$.
- **Laplacian Spectrum**: Key structural characteristics of a graph, such as its expansion properties and bottlenecks, are closely related to the eigenvalues of its normalized Laplacian matrix $\mathbf{L}$. As a prominent example, the second-

smallest eigenvalue is related to the Cheeger constant, a value that measures the difficulty of partitioning the graph into separate components.

We remark that the resistance-distance (RD) matrix of a given graph G, denoted by R, can be computed by the generalized inverse, $\mathbf{L}^*$, of the Laplacian matrix of the underlying graph G. More precisely, each entry $R_{ij} = \mathbf{L}^*_{ii} + \mathbf{L}^*_{jj} - \mathbf{L}^*_{ij} - \mathbf{L}^*_{ji}$ [22]. Note that the Laplacian matrix of a graph is singular, and it thus has no usual inverse; however, the generalized inverse of a singular matrix can be found. Moreover, in this study, we exploit the normalized Laplacian matrix of a graph which can reflect more structural properties of the underlying graph [23].

Here we applied the spectral method to the resistance-equivalent graph pairs (see Figure 6). The spectral method successfully differentiated them. Both the adjacency and Laplacian spectra exhibited distinct eigenvalue patterns between the resistance-equivalent graph pairs.

1) Strengths of Spectral Representation: Our research journey can be summarized as follows:

In scenarios involving graph pairs that possess identical node counts and equal numbers of edges, and—yet are non-isomorphic—resistance distance has shown effectiveness in distinguishing them. Our spectral-based method was also successful in these cases.

This contrast highlights the stronger expressive power of the spectral approach over resistance distance. Notably, it also draws from well-established principles in spectral graph theory. The key advantages of this approach include:

- **Solid Theoretical Foundation:** Eigenvalues have profound theoretical meaning and rich properties in algebraic graph theory.
- **Stronger Expressive Power:** It can successfully handle specific challenging graphs that even resistance distance fails to distinguish.
- **Extensibility:** Beyond eigenvalues, their corresponding eigenvectors also contain rich topological information, pointing to ways to further enhance graph representations in future work.

IV. Distinguishing Resistance-Equivalent Graphs: A Spectral Approach

We provide counterexamples demonstrating the failure of resistance distance to distinguish specific graph pairs, while our spectral approach succeeds.

A. Challenge of Resistance-Equivalent Graphs

Resistance-equivalent graphs are defined as structurally different (i.e., non-isomorphic) graphs that nevertheless share the same *resistance spectrum* — the multiset of all pairwise resistance distances between nodes in the graph.

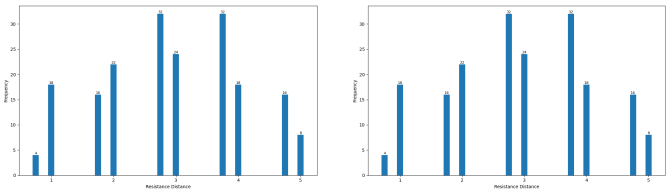
To make this limitation more concrete, we present a pair of non-isomorphic graphs—denoted G_1 and G_2—in Fig. 6. Despite having different structures, the two graphs are nearly indistinguishable based on many commonly used graph features:

- The same number of nodes ($|V|$)
- The same number of edges ($|E|$)
- An identical degree sequence

Because these basic properties align perfectly, distinguishing G_1 from G_2 is especially difficult for resistance-based methods, graph algorithms, and machine-learning models. Moreover, the resistance spectra of the two graphs are identical. In other words, the resistance spectrum $\mathsf{RS}(G)$, which collects all pairwise resistance distances, is identical for both graphs:

$$\mathsf{RS}(G_1) = \mathsf{RS}(G_2) = [\![\tfrac{3}{4}]\!]^4 [\![1]\!]^{18} [\![\tfrac{7}{4}]\!]^{16} [\![2]\!]^{22} [\![\tfrac{11}{4}]\!]^{32} [\![3]\!]^{24} [\![\tfrac{15}{4}]\!]^{32} [\![4]\!]^{18} [\![\tfrac{19}{4}]\!]^{16} [\![5]\!]^8$$

Here, $[\![x]\!]^n$ indicates that the resistance value x appears n times.



(a) G_1: RD Distribution (b) G_2: RD Distribution

Fig. 8: Resistance distance distributions for the resistance-equivalent graph pair G_1 and G_2 shown in Fig. 6. In each histogram, the x-axis represents the resistance distance, and the y-axis indicates its frequency. The two histograms are visually identical, confirming that both graphs share the same resistance spectrum.

To visualize, we compare the resistance distance distributions of G_1 and G_2 using histograms, as shown in Fig. 8. The resistance distance distributions for G_1 and G_2 are identical, although the two graphs are not isomorphic. This shows that the resistance spectrum alone cannot distinguish G_1 from G_2.

In short, neither basic structural features nor resistance distances can distinguish between these types of graphs, revealing the limitations of representation methods that rely on resistance distances. To solve the problem of distinguishing between these difficult-to-distinguish graphs, our work utilizes the spectral properties of graphs—their eigenvalues and eigenvectors—which can capture subtle structural differences that RD overlooks.

B. Spectral-based Representation

In spectral graph theory, a graph's entire structure is encoded in matrices such as the Adjacency Matrix (A) and the Laplacian Matrix (L). The eigenvalues and eigenvectors of these matrices—their *spectral properties*—reveal structural information that local or path-based methods often fail to capture.

This approach is grounded in solid theoretical foundations. The eigenvalues of a graph's Laplacian matrix, for example, are known to be tied to key properties such as connectivity and how easily the graph can be partitioned. Therefore, spectral features are not arbitrary numbers; they are direct reflections of a graph's topology. Even if two graphs have the same resistance spectrum, they will have different adjacency or

Laplacian spectra. These spectral features can therefore act as a more reliable "graph fingerprint."

Specifically, we employ two kinds of spectral features to build our representation:

Eigenvalues: For a given graph, the set of eigenvalues from one of its matrices (e.g., Adjacency or Laplacian), known as its spectrum, provides a global and permutation-invariant description. We leverage this property to compare representations between graphs.

Eigenvectors: By contrast, eigenvectors provide information at the node level. Each eigenvector assigns a scalar to every node, which can be interpreted as its role in a particular structural pattern of the graph. We use the eigenvectors from the normalized Laplacian matrix to create a low-dimensional embedding for each node. The resulting embeddings capture a node's topological role and position within the graph, thereby yielding node-level representations.

In the following subsection, we empirically demonstrate that our spectral approach, using either eigenvalues or eigenvectors, distinguishes graphs that are resistance-equivalent, which cannot be distinguished by resistance distance alone.

C. Experimental Validation

In the previous section, we argued that a spectral approach could offer a more discriminative graph representation than resistance distance. We now empirically validate this claim by analyzing the pair of resistance-equivalent graphs, G_1 and G_2, from Fig. 6. We demonstrate our spectral approach at both the graph level, using eigenvalues, and at the node level, using eigenvectors.

1) Eigenvalue Comparison: We conducted a comparative spectral analysis of the graph-level properties between G_1 and G_2. Specifically, we calculated the largest k eigenvalues of each graph's adjacency matrix and the lowest k non-zero eigenvalues from their Laplacian matrix, setting $k = 10$ for illustrative purposes. Given that the smallest eigenvalue of any Laplacian matrix is invariably zero, we excluded it and instead presented the following 10 smallest eigenvalues (ranging from λ_2 to λ_{11}). Tables I and II summarize these computed values, clearly revealing numerical differences between their spectral features despite their equivalence in resistance distance. We represent each eigenvalue by λ_i, where the subscript i denotes its sorted position within the spectrum.

TABLE I: Largest k eigenvalues of the adjacency matrix for G_1 and G_2 ($k = 10$). Values in **bold** indicate eigenvalues that differ between G_1 and G_2.

Graph	λ_1	λ_2	λ_3	λ_4	λ_5
G_1	2.819882	**1.806991**	**1.796406**	**1.461515**	1.000000
G_2	2.819869	**1.847759**	**1.732051**	**1.508025**	1.000000

Graph	λ_6	λ_7	λ_8	λ_9	λ_{10}
G_1	1.000000	**1.000000**	**0.597844**	**0.250062**	$\mathbf{-2.4 \times 10^{-16}}$
G_2	1.000000	**0.879886**	**0.765367**	$\mathbf{8.4 \times 10^{-16}}$	$\mathbf{4.0 \times 10^{-16}}$

In Table I, although the maximum eigenvalues of the two graphs are the same, the second, third, and fourth eigenvalues show differences. Similarly, in Table II, the Laplacian spectra of G_1 and G_2 are distinguishable.

TABLE II: Smallest non-zero k eigenvalues of the Laplacian matrix for G_1 and G_2 ($k = 10$). Values in **bold** indicate eigenvalues that differ between G_1 and G_2.

Graph	λ_2	λ_3	λ_4	λ_5	λ_6
G_1	**0.118241**	**0.121288**	**0.170696**	0.292893	0.292893
G_2	**0.111926**	**0.133975**	**0.160879**	0.292893	0.292893

Graph	λ_7	λ_8	λ_9	λ_{10}	λ_{11}
G_1	**0.417895**	**0.608115**	**0.857869**	1.000000	1.000000
G_2	**0.456055**	**0.540299**	**1.000000**	1.000000	1.000000

These numerical differences provide clear evidence that, although these graphs are resistance-cospectral, they are not cospectral in terms of their adjacency matrices and Laplacian matrices. Therefore, the graph spectral features constructed from the eigenvalues successfully distinguish the two non-isomorphic graphs.

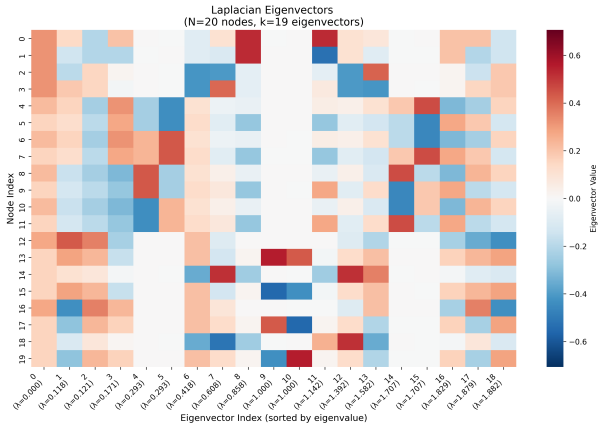
2) Eigenvector Representation: Next, we explore representations at the node level. We use the eigenvectors of the normalized Laplacian matrix to construct an embedding for each node. Specifically, we take the eigenvectors corresponding to the k smallest eigenvalues (from λ_1 to λ_k) to create a k-dimensional feature vector for every node in both G_1 and G_2. In our analysis, we set $k = 20$.

To visually compare the overall structure of these node embeddings, we plot the resulting $N \times k$ node-feature matrices as heatmaps, as shown in Fig. 9. To take a closer look at the behavior of individual eigenvectors, we also plot the values of the first few eigenvectors as line profiles in Fig. 10. Comparing Eigenvector 3 (corresponding to λ_3) in both graphs, for instance, reveals apparent differences in their value distributions and wave patterns. Eigenvector 3 for G_1 peaks at nodes 12 and 16, whereas Eigenvector 3 for G_2 peaks at nodes 4 and 6. These subtle distinctions accumulate to create distinct overall representations.

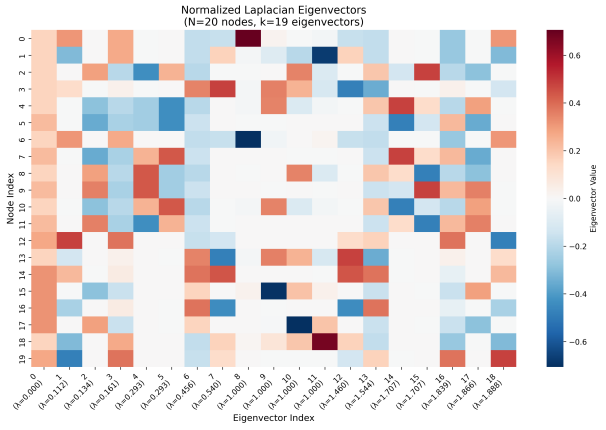
These findings demonstrate that node representations constructed from eigenvectors are successful on two fronts. They effectively distinguish the resistance-equivalent graphs, and they do so by capturing the specific internal structural patterns that RD features miss.

V. Evaluating Spectral Representations on Benchmark Datasets

In the previous section IV, we demonstrated the expressive power of our spectral-based representation, showing that this representation can successfully distinguish resistance-equivalent graphs that pose a challenge for other methods. Having established its theoretical advantages, this section aims to evaluate the practical effectiveness of this approach on downstream machine learning tasks. To this end, we integrate our spectral features into the Graphormer architecture, creating the Spectral-Graphormer model. We then evaluate its performance on public benchmark datasets for graph regression (ZINC) and graph classification (OGBG-PPA).

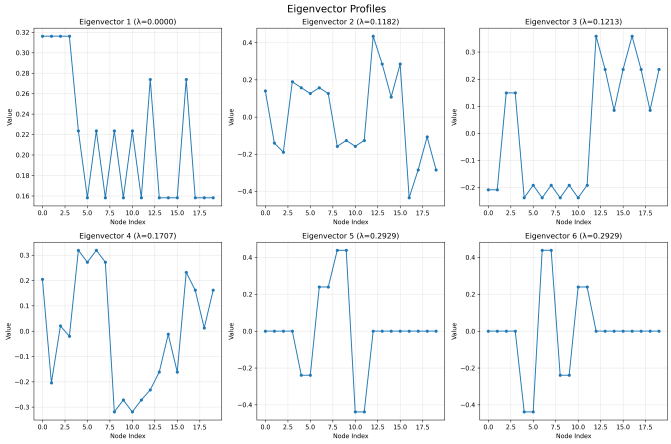


(a) Node embeddings of G_1

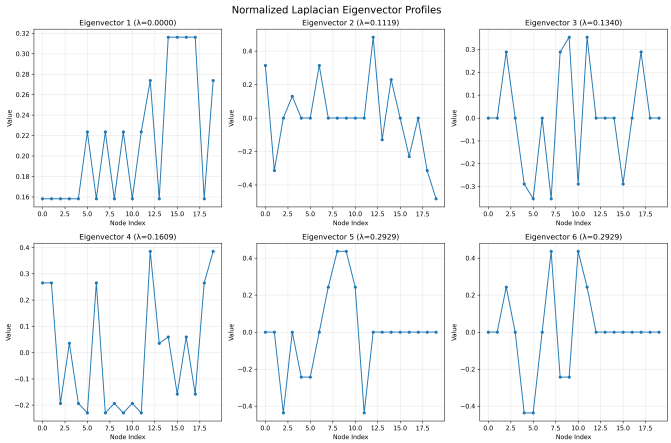


(b) Node embeddings of G_2

Fig. 9: Heatmaps of node embeddings for G_1 and G_2, constructed using the first 20 eigenvectors of the normalized Laplacian. In each heatmap, the y-axis represents the node index, while the x-axis corresponds to the eigenvector index, sorted by eigenvalue. The contrasting patterns in (a) and (b) highlight node-level structural differences between the two graphs.



(a) First 6 eigenvectors of G_1



(b) First 6 eigenvectors of G_2

Fig. 10: Line plots of the first few eigenvectors for G_1 and G_2. Each plot shows the eigenvector's values (y-axis) across all node indices (x-axis). The differences in value distributions and oscillation patterns underscore the structural distinctions captured by spectral features.

Through these experiments, we aim to validate the effectiveness of our proposed spectral method against strong baseline models. Crucially, we also aim to demonstrate its superior scalability and computational efficiency on large-scale graphs.

A. Experimental Setup

1) Proposed Model: We propose the Spectral-Graphormer, a model designed to evaluate the spectral features introduced in Section IV-C2. Specifically, we encode these spectral features into the Graphormer framework [4], a Transformer-like architecture for graphs.

We select Graphormer as our base model because it incorporates several structural encoding schemes, including Centrality Encoding (based on node degrees), Spatial Encoding (based on the shortest path distance), and Edge Encoding. We aim to provide a more powerful representation by augmenting this architecture with spectral features, as spectra are grounded in well-established graph theory and capture rich topological information.

2) Baseline Models: We evaluate our model against representative models from three prominent families of GNNs: Graph Transformers, Message Passing Neural Networks (MPNNs), and Subgraph GNNs.

a) Graph Transformer: Graphormer and Graphormer-GD: We include two Graph Transformer baselines. The original **Graphormer** [4] serves as a direct baseline to evaluate the performance gain from our proposed spectral features. **Graphormer-GD** [5], a model that emphasizes the importance of structural information, focuses on distance metrics—specifically, the Shortest Path Distance (SPD) and Resistance Distance (RD). For SPD, its approach adopts the same encoding method as the original Graphormer [4]. Their approach applies Gaussian Basis kernels to encode RD, which can take non-integer values. The outputs from different kernels are then concatenated and processed through a two-layer MLP to produce the final feature.

b) Message Passing Neural Network: PNA: As a representative of MPNNs, we select Principal Neighborhood Aggregation (PNA) [7]. PNA enhances traditional GNNs by combining multiple aggregators (e.g., mean, max, min) with degree-scalers, allowing the model to capture the neighborhood feature distribution better.

c) Subgraph GNN: SUN: From the Subgraph GNN family, we select the Subgraph Union Network (SUN) [6]. SUN enhances graph representation by applying a GNN over a collection of subgraphs sampled for each node, thereby capturing a richer set of local topological structures. This node-based subgraph selection policy, which extracts an h-hop neighborhood for each node, is highly effective for small molecular graphs, such as those in ZINC. However, this approach leads to prohibitive memory consumption on large-graph datasets such as OGBG-PPA, making it computationally infeasible for our experiments on that benchmark.

3) Hardware and Software Environment: We conducted our experiments on a server equipped with an Intel Xeon Gold 6154 CPU and four NVIDIA Tesla V100-SXM2 GPUs, each with 32 GB of memory. The standard configuration for our experiments utilized 240 GB of system RAM. However, to accommodate the substantial memory requirements of the Graphormer-GD baseline on the OGBG-PPA dataset, we ran this specific experiment on a high-memory configuration with 480 GB of RAM.

B. Graph Regression on ZINC

1) ZINC Dataset: We conducted experiments on the public ZINC benchmark dataset. The ZINC is a large-scale collection of commercially available chemical compounds curated for virtual screening in chemical biology and drug discovery [24].

Among its commonly used subsets, ZINC-250K is a large-scale collection containing approximately 250,000 molecular graphs and is sometimes referred to in the literature as *ZINC-full*. Correspondingly, ZINC-12K is a widely used benchmark subset consisting of 12,000 molecular graphs and is commonly adopted for evaluating GNNs [25]. The node features of each molecular graph correspond to heavy atom types, while edge features represent chemical bond types. The task is a graph regression task aimed at predicting the constrained solubility of each molecule, and the evaluation metric is the Mean Absolute Error (MAE).

2) Model Architecture: Our Spectral-Graphormer used the $k = 20$ eigenvectors of the normalized Laplacian, as described in Section IV-C2. For a fair comparison on the ZINC-12K, all models adhered to the parameter budgets established by Dwivedi et al. [25]. Specifically, for the Transformer-based models, we adopted the consistent hyperparameter setup from the official Graphormer-GD implementation. This architecture comprised a slim 12-layer Transformer, each with eight attention heads, a hidden dimension of 80, and a feed-forward network (FFN) inner-layer hidden dimension of 80. For regularization, we applied a dropout rate of 0.1 to the FFN, attention, and embedding layers. The 0.1 rate for embedding dropout, in particular, follows standard practice from prominent Transformer models [26], [27].

3) Training Details: All Transformer-based models were trained for a maximum of 10,000 epochs with a batch size of 256. We used the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, and a gradient clipping norm of 5.0. The learning rate schedule followed a linear warm-up scheme with a peak learning rate of 5×10^{-4} and 60,000 warm-up steps. The total training steps and hardware were adjusted for each dataset. For ZINC-250K, models were trained for 800,000 steps across 4 NVIDIA Tesla V100 GPUs. For ZINC-12K, they were trained for 600,000 steps on a single GPU.

We reproduced PNA and SUN using their official implementations with the authors' default hyperparameter settings on ZINC.

4) Results and Analysis: The performance of all models on both the ZINC-12K (subset) and ZINC-250K (full) datasets is shown in Table III.

TABLE III: Graph regression performance on ZINC datasets. The table reports the test MAE at the best validation epoch (lower is better).

Method	**Model**	**Test MAE (↓)**	
		ZINC-12K	**ZINC-250K**
MPNNs	PNA [7]	0.1554	–
Subgraph GNNs	SUN [6]	0.0773	–
Graph Transformers	Graphormer [4]	0.1600	0.0536
	Graphormer-GD [5]	0.0930	0.0365
	Spectral-Graphormer	**0.1267**	**0.0380**

On the ZINC-12K, the results show that SUN (0.0773 MAE) achieves the best performance on this task, indicating the strength of the subgraph-based GNN architecture. Our comparison focus, however, lies with the Graph Transformer family. Within this group, Spectral-Graphormer (0.1267 MAE) outperforms the original Graphormer (0.1600 MAE). This result validates the effectiveness of adding spectral features to this architectural approach.

On the large-scale ZINC-250K benchmark, our Spectral-Graphormer (0.0380 MAE) significantly improves upon the original Graphormer baseline (0.0536 MAE) and performs on par with the Graphormer-GD baseline (0.0365 MAE). A similar trend is observed in the smaller ZINC-12K, where our model also demonstrates that spectral features are effective, even without model-specific tuning. In the following sections, we focus our detailed analysis on the ZINC-250K dataset.

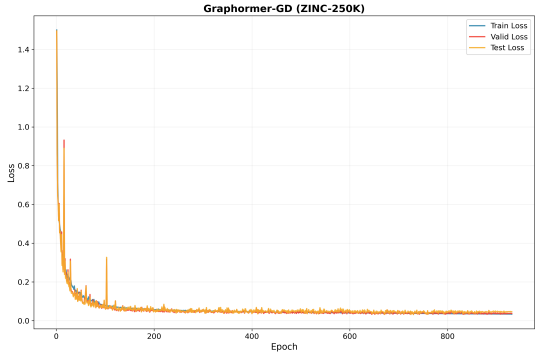
TABLE IV: MAE values on ZINC-250K at the best validation epoch for each model. A smaller gap between training and validation/test MAE indicates better generalization.

Model	**Train MAE**	**Valid MAE**	**Test MAE**
Graphormer [4]	0.0718	0.0591	0.0536
Graphormer-GD [5]	0.0402	0.0329	0.0365
Spectral-Graphormer	**0.0407**	**0.0379**	**0.0380**

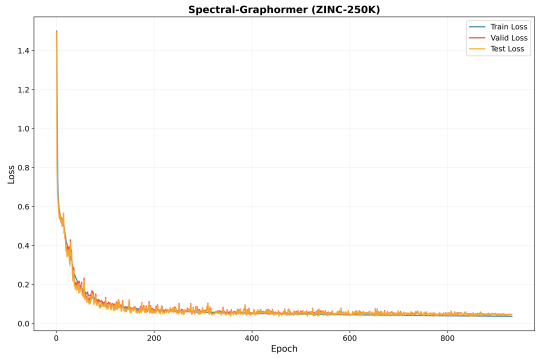
To further analyze generalization performance, we compare the loss values of the models at their respective best validation

epochs in Table IV. While Graphormer-GD and Spectral-Graphormer have similar training losses, our model exhibits a smaller generalization gap (i.e., the difference between training and validation/test losses), suggesting it has better generalization capabilities.

The primary goal of this experiment was to validate the viability and initial effectiveness of spectral features in GNNs. Therefore, we directly adopted the hyperparameter settings from Graphormer-GD without additional tuning for our model. This competitive result indicates that spectral features are a powerful topological feature, with further gains possible through future optimization for Spectral-Graphormer.



(a) Graphormer-GD



(b) Spectral-Graphormer

Fig. 11: Training loss curves on ZINC-250K for Graphormer-GD and Spectral-Graphormer. The comparison focuses on Graphormer-GD, the strongest baseline, while other models are omitted for clarity.

5) Convergence Analysis: We visualize the loss curves for the Graphormer-GD and Spectral-Graphormer models in Figure 11. Both models converge smoothly. Furthermore, the validation and test loss trajectories closely follow the training loss, indicating no significant signs of overfitting. We focus on Graphormer-GD in particular, as it is the most competitive baseline among the models. While other models, such as Graphormer, are also included in our performance comparison (Table III), we omit their loss curves here to maintain clarity and emphasize the most relevant comparisons.

C. Graph Classification on OGBG-PPA

1) OGBG-PPA Dataset: We further evaluate our model on the OGBG-PPA (Open Graph Benchmark—Protein–Protein Association) dataset [28], a large-scale benchmark for graph classification. The task is a multi-class classification problem to predict which of the 37 biological taxonomic groups a given protein-protein association network belongs to. Performance is measured by accuracy.

The OGBG-PPA dataset contains 158,100 graphs, with an average of 243.4 nodes (proteins) and 2,261.1 edges (associations) per graph. This dataset does not include node features but provides 7-dimensional edge features with values between 0 and 1, representing the confidence scores of protein-protein interactions.

The dataset employs a challenging *species split* strategy, which differs from a conventional random split, in which graphs are randomly assigned to the training, validation, and test sets. Specifically, the graphs in the validation and test sets belong to species not seen during training; however, they still fall under one of the 37 categories. This setup means that the training, validation, and testing data are from different distributions. This split evaluates a model's generalization ability on unseen species; at the same time, it poses a significant challenge.

2) Model Architecture: For the OGBG-PPA experiments, our Spectral-Graphormer model utilized the first $k = 20$ eigenvectors of the normalized Laplacian, as described in Section IV-C2. All Transformer-based models, including our Spectral-Graphormer and the baselines, were configured with a consistent set of architectural hyperparameters for a fair comparison. The architecture consisted of a 5-layer Transformer with eight attention heads per layer, where the hidden dimension and the FFN inner-layer dimension were set to 256. For regularization, we applied dropout, attention dropout, and activation dropout, each at 0.3, along with a stochastic depth (drop-path) probability of 0.3.

For the PNA model, the network architecture consisted of 4 layers with hidden and output dimensions of 80, using a mean readout function. We employed a combination of "mean," "max," "min," and "std" aggregators with "identity," "amplification," and "attenuation" scalers, and used one pre-transformation and one post-transformation layer.

3) Training Details: All Transformer-based models were trained using the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 1 \times 10^{-8}$, a weight decay of 0.05, and gradient clipping with a norm of 5.0. The learning rate schedule included a linear warm-up phase for the first 40,000 steps to a peak learning rate 3×10^{-4}. Following the warm-up, the learning rate linearly decayed to a final rate of 1×10^{-5}. We used a batch size of 32 and trained the models for 800,000 steps on 4 NVIDIA Tesla V100 GPUs for this dataset.

For the PNA model, we adapted the hyperparameter configuration from the authors' official implementation for the ZINC and OGBG-molhiv datasets. Our setup included a batch size of 32, an initial learning rate of 0.01 with a reduction factor of 0.5, patience of 20 epochs, a minimum learning rate of 1×10^{-4}, and a weight decay of 3×10^{-6}.

4) Results and Analysis: We evaluated all models on OGBG-PPA, with the exception of SUN. As described in Section V-A2, SUN's subgraph sampling strategy is memory-intensive and becomes prohibitive for the large graphs in

OGBG-PPA. In contrast, our proposed Spectral-Graphormer can efficiently process these graphs, demonstrating its practical advantage in this challenging task. The results are presented in Table V.

TABLE V: Graph classification performance on the OGBG-PPA dataset. The table reports the accuracy (%) at the same epoch.

Method	**Model**	**Accuracy (%)**		
		Training	Validation	**Test**
MPNNs	PNA [7]	64.9	52.5	58.2
Graph Transformers	Graphormer [4]	54.8	35.0	39.7
	Graphormer-GD [5]	49.2	37.0	42.1
	Spectral-Graphormer	**82.9**	**50.6**	**53.9**

The results indicate that our Spectral-Graphormer substantially outperforms both Graphormer and Graphormer-GD on the validation (50.6%) and test (53.9%) sets. Notably, our model achieves performance comparable to strong MPNN baselines such as PNA (58.2%). However, we also observed a large gap between the training accuracy (82.9%) and the validation/test scores, indicating that our model experienced more pronounced overfitting.

The overfitting prompted us to attempt to reduce the model's complexity. We reduced the hidden and FFN sizes from 256 to 128 while keeping all other settings unchanged. After this adjustment, the train/validation/test accuracies became 72.9%, 45.9%, and 51.2%, respectively. As expected, the gap between the training and validation/test sets shrank, confirming that reducing model complexity can improve performance on challenging datasets like OGBG-PPA, especially with its tough species-split.

5) Efficiency and Scalability Analysis: On the OGBG-PPA dataset, which features larger graphs, we observed that Graphormer-GD demands more memory. Its resistance distance feature requires computing the distance between every pair of nodes, yielding a dense $N \times N$ matrix for each graph (where N is the number of nodes). This resulted in a peak memory consumption of 335 GB in our experiments.

In contrast, our Spectral-Graphormer computes the *first* k eigenvectors of the normalized Laplacian for each graph, producing an additional $N \times k$ matrix. In our experiments on the OGBG-PPA dataset, our model's peak memory usage was only 177 GB—substantially lower than Graphormer-GD [5].

Our approach is substantially more efficient in both computation and storage because we only compute the top k eigenvectors, where k is typically much smaller than the number of nodes N ($k \ll N$). This efficiency is particularly important for datasets such as OGBG-PPA, where the average number of nodes is 243.4. The validity of using a small k is well-established in spectral graph theory, as the first few eigenvectors of the normalized Laplacian, corresponding to the smallest eigenvalues, are often sufficient to capture the global structural information of a graph [25], [29], [30].

While this efficiency advantage may be less pronounced on smaller molecular graphs, such as those in ZINC, the superior scalability of our method becomes evident in datasets with a larger number of nodes, such as OGBG-PPA.

VI. Conclusion

In this study, we propose Spectral-Graphormer, a graph neural network model based on spectral features, to address the challenges of distinguishing non-isomorphic graphs that existing graph-representation methods cannot. Our goal is to overcome the limitations of distinguishing ability in state-of-the-art models, such as Graphormer-GD [5].

The primary contributions of this work lie in its expressive power to distinguish graphs. First, graph pairs that can be distinguished by resistance-distance features (as used in Graphormer-GD) can also be distinguished by the eigenvalues or eigenvectors of the (normalized) Laplacian employed in our method. Second, our spectral approach successfully differentiates resistance-equivalent graphs—a class of non-isomorphic graphs for which resistance distances fail. These graphs share identical resistance spectra despite being structurally non-isomorphic. We use Laplacian spectral features to capture the subtle topological differences between them, confirming the superior distinguishing ability of our approach.

In practical applications, we evaluated Spectral-Graphormer on two public benchmark datasets: ZINC (molecular property prediction) and OGBG-PPA (protein-protein association classification). The results show that our model's performance is comparable to that of state-of-the-art models and, in some cases, outperforms them. These results indicate that our method is competitive on real-world benchmarks.

A key practical advantage of our method is its efficiency and scalability for large-scale graphs. Unlike Graphormer-GD, which requires a memory-intensive $N \times N$ distance matrix, our method computes only the first k eigenvectors, resulting in a much smaller $N \times k$ feature matrix (where $k \ll N$). This significantly reduces memory consumption, making our approach more feasible for large-scale datasets such as OGBG-PPA—precisely where distance-based models (e.g., Graphormer-GD) and subgraph-based GNNs (e.g., SUN [6]) encounter severe memory bottlenecks.

We did not conduct an extensive hyperparameter search. Future work could fine-tune Spectral-Graphormer to potentially achieve further performance gains. Furthermore, this work validated the effectiveness of incorporating spectral features into the Graph Transformer architecture. A natural extension would be to investigate the integration of these spectral features as a general structural encoding scheme into other GNN architectures.

References

[1] W. Ju, Z. Fang, Y. Gu, Z. Liu, Q. Long, Z. Qiao, Y. Qin, J. Shen, F. Sun, Z. Xiao *et al.*, "A comprehensive survey on deep graph representation learning," *Neural Networks*, p. 106207, 2024.

[2] R. Das and M. Soylu, "A key review on graph data science: The power of graphs in scientific studies," *Chemometrics and Intelligent Laboratory Systems*, vol. 240, p. 104896, 2023.

[3] I. Bronštejn and K. Semendjaev, *Handbook of mathematics*. Springer, 2013.

[4] Z. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform bad for graph representation?" in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.

[5] B. Zhang, S. Luo, L. Wang, and D. He, "Rethinking the expressive power of GNNs via graph biconnectivity," in *International Conference on Learning Representations (ICLR)*, 2023.

[6] F. Frasca, B. Bevilacqua, M. Bronstein, and H. Maron, "Understanding and extending subgraph gnns by rethinking their symmetries," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 376–31 390, 2022.

[7] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 260–13 271, 2020.

[8] B. Weisfeiler and A. Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *NTI, Series*, vol. 2, no. 9, pp. 12–16, 1968.

[9] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are GNNs?" in *International Conference on Learning Representations (ICLR)*, 2019.

[10] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4602–4609.

[11] S.-A. Xu, H. Zhou, and X.-F. Pan, "A method for constructing graphs with the same resistance spectrum," *Discrete Mathematics*, vol. 348, no. 2, p. 114284, 2025.

[12] P. G. Doyle and J. L. Snell, *Random walks and electric networks*. American Mathematical Society, 1984, vol. 22.

[13] D. J. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993.

[14] E. F. Sanmartín, S. Damrich, and F. Hamprecht, "The algebraic path problem for graph metrics," in *International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 19 178–19 204.

[15] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 1998, vol. 184.

[16] D. J. Klein, "Resistance-distance sum rules," *Croatica Chemica Acta*, vol. 75, no. 2, pp. 633–649, 2002.

[17] N. Biggs, *Algebraic graph theory*. Cambridge University Press, 1993.

[18] D. Cvetković, M. Doob, and H. Sachs, *An introduction to the theory of graph spectra*. Cambridge University Press, 2009.

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

[20] D. A. Spielman, "Algorithms, graph theory, and linear equations in laplacian matrices," in *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010)*, vol. 4, 2010, pp. 2698–2722.

[21] C. Godsil and G. Royle, *Algebraic graph theory*. Springer, 2001.

[22] W. Xiao and I. Gutman, "Relations between resistance and laplacian matrices and their applications," *MATCDY*, vol. 51, pp. 119–127, 2004.

[23] H. Chen and Z. Fuji, "Resistance distance and the normalized laplacian spectrum," *Discrete Applied Mathematics*, vol. 155, pp. 654–661, 2006.

[24] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman, "ZINC: A free tool to discover chemistry for biology," *Journal of Chemical Information and Modeling*, vol. 52, no. 7, pp. 1757–1768, 2012.

[25] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *Journal of Machine Learning Research*, vol. 24, no. 43, pp. 1–48, 2023.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[28] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 22 118–22 133.

[29] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[30] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *International Conference on Learning Representations (ICLR)*, 2014.