

GRAPHFRACTALNET: A FRACTAL-INSPIRED SPARSE TRANSFORMER FOR ULTRA-SCALABLE GRAPH REPRESENTATION LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) and Transformer-based architectures have achieved strong performance in graph representation learning, yet they often struggle with scalability, over-smoothing, and limited expressiveness on complex topological patterns. We propose GraphFractalNet, a novel framework that integrates spectral embeddings, dynamic graph rewiring, and a fractal attention mechanism to capture both global and hierarchical self-similar structures in graphs. By leveraging truncated spectral bases, GraphFractalNet provides topology-aware node embeddings, while the rewiring module adaptively optimizes edge connectivity to improve information flow and sparsity. The fractal attention layer further constrains attention to recursively clustered subgraphs, enabling sub-quadratic complexity of $O(N \log \log N)$ per layer while retaining expressive power beyond k -Weisfeiler–Lehman tests. Theoretically, we establish generalization bounds under spectral Rademacher complexity and prove that GraphFractalNet is strictly more expressive than standard message-passing GNNs. Empirical results show that GraphFractalNet delivers state-of-the-art performance on both molecular property prediction and large-scale node classification tasks, consistently improving accuracy and scalability. Comprehensive ablation studies underscore the critical roles of spectral embeddings, dynamic rewiring, and fractal attention, each contributing to the model’s effectiveness and efficiency. Overall, GraphFractalNet emerges as a principled and scalable architecture that seamlessly integrates spectral techniques with Transformer-inspired designs for graph learning.

1 INTRODUCTION

Representation learning on graphs has rapidly matured due to the convergence of message-passing Graph Neural Networks (GNNs) and Transformer-style attention architectures. Classic GNNs such as GCN Kipf (2017), GIN Xu et al. (2019), and GraphSAGE Hamilton et al. (2017) excel at encoding local neighborhood information through iterative aggregation. However, they often struggle to model long-range dependencies, encounter over-smoothing at depth, and lack the expressiveness required to distinguish graphs that differ only in global or higher-order structures.

To address the limitations of traditional GNNs, graph Transformers have incorporated structural encodings such as shortest-path distances, edge attributes, and centrality measures, which significantly improve performance on tasks like molecular property prediction and graph classification. Recent studies Ying et al. (2021) have shown that incorporating structural encodings is essential for the success of Transformers on graph-related tasks. Similarly, the Topology-Informed Graph Transformer Choi et al. (2024) introduces topological positional embeddings based on cycles and community structures, allowing the model to better capture complex graph patterns. Despite these advances, most graph Transformers still rely on dense quadratic attention mechanisms ($O(N^2)$), leading to high computational costs, and their structural encodings remain largely static and hand-crafted, limiting adaptability across diverse graph domains.

In parallel, spectral GNNs exploit eigen-decompositions of the Laplacian to encode rich global structure. Classical methods such as ChebNet and Laplacian positional encodings employ truncated eigenbases to inject spectral information into node embeddings. Enhancing Spectral GNNs: From

Topology and Perturbation Qin et al. (2025) demonstrates that perturbed Laplacians sharpen eigenvalue separability, thereby improving discriminative capacity, while Spatio-Spectral GNNs Geisler et al. (2024) integrate spectral filtering with spatial message passing to mitigate over-squashing and achieve expressiveness beyond the 1-WL test. Nevertheless, spectral approaches generally lack adaptive attention mechanisms, limiting their ability to capture graph heterogeneity and dynamically emphasize task-relevant structural patterns.

Efforts to balance efficiency with structural awareness have led to several promising directions. SFi-Former Li et al. (2025) casts attention as a sparse flow-selection process, enabling long-range interactions with reduced complexity, while ReHub Borreda et al. (2024) employs a hub-spoke attention design that scales linearly with the number of nodes. Complementary approaches, such as Graph Sparse Training (GST) Zhang et al. (2024), focus on dynamic edge pruning to improve scalability and robustness. Although these models demonstrate clear benefits in sparsity and adaptability, they often neglect the incorporation of spectral priors and hierarchical subgraph structure, which are critical for capturing the full spectrum of graph dependencies across multiple scales.

Graphs in many domains exhibit hierarchical and self-similar patterns, where local connectivity is nested within larger organizational structures that repeat across scales. Capturing these patterns is critical for effective representation learning, yet remains a persistent challenge for current approaches. Conventional Graph Transformers rely on dense attention and static structural encodings, making them computationally demanding and insufficiently adaptive to diverse graph topologies. Spectral methods offer principled topology-aware embeddings through Laplacian eigenvectors, but they are often fixed and not directly integrated with adaptive attention or connectivity control. Sparse Transformers and rewiring-based approaches improve scalability, but they frequently overlook multi-scale dependencies and hierarchical structural cues, leading to limited expressiveness.

To overcome these challenges, we introduce GraphFractalNet, a unified architecture that combines spectral encoding, dynamic rewiring, and fractal attention in a Transformer framework. The spectral encoder computes truncated Laplacian eigenvectors to provide multi-scale node and edge embeddings, grounding the model in the intrinsic geometry of the graph. A dynamic rewiring module adaptively updates graph connectivity at each layer by selecting structurally relevant edges while preserving global connectivity. On top of this, a fractal attention mechanism organizes attention hierarchically across clusters obtained through spectral partitioning, enabling efficient yet expressive aggregation with per-layer complexity reduced to $O(N \log \log N)$. By integrating these components, GraphFractalNet achieves both scalability and expressiveness, effectively capturing global structure, local interactions, and hierarchical self-similarity in a single coherent model.

The main highlights of this paper are outlined as follows:

1. We design an end-to-end framework that combines Laplacian-based spectral embeddings, adaptive graph rewiring, and fractal-inspired hierarchical sparse attention to capture rich structural patterns in graphs.
2. By exploiting hierarchical sparsity, GraphFractalNet achieves sub-quadratic per-layer complexity $O(N \log \log N)$, enabling efficient training on graphs with tens of thousands of nodes while offering provable bounds on efficiency.
3. We prove that GraphFractalNet can distinguish graphs strictly beyond the k -WL hierarchy ($k \geq 2$) under mild spectral conditions, and we derive generalization bounds using Rademacher complexity that scale favorably with spectral dimension, sparsity, and depth.
4. GraphFractalNet is extensively evaluated on diverse graph-level prediction and node classification tasks, spanning both medium- and large-scale benchmarks. The results consistently show superior accuracy and robustness compared to strong baselines.

2 RELATED WORK

2.1 GRAPH TRANSFORMERS & STRUCTURAL ENCODINGS

Graph Transformers adapt the self-attention paradigm from sequential data processing Vaswani et al. (2017) to handle graph-structured inputs, providing robust alternatives to conventional GNNs Dwivedi & Bresson (2020). Initial variants of these models confine attention computations to prox-

imal node neighborhoods, effectively operating as attention-augmented message-passing schemes Joshi (2025); Bronstein et al. (2021), and rely on positional embeddings derived from the eigenvectors of the graph Laplacian Dwivedi et al. (2023a). More advanced Graph Transformers, however, adopt fully global attention strategies, enabling every node to interact directly with all others Mialon et al. (2021); Kreuzer et al. (2021). While this design transcends the neighborhood-bound constraints of traditional GNNs Alon & Yahav (2021), it incurs a substantial rise in computational demands.

Transformer-based graph models integrate structural encodings to improve expressivity and performance. GraphGPS Rampásek et al. (2022) unifies message passing with global attention and various positional encodings. Graphormer variants DeepGraph Zhao et al. (2023), Structure-Aware Transformer (SAT) Chen et al. (2022), and Simple Path Structural Encoding (SPSE) Airale et al. (2025) introduce positional encodings based on paths or subgraphs to capture richer structure. Hierarchical Distance Structural Encoding (HDSE) Luo et al. (2024) encodes multi-level distances, advancing structural awareness. However, these models often rely on dense attention (quadratic in node count), limiting scalability.

2.2 SPECTRAL GNNs AND POSITIONAL ENCODING

Graph Transformer architectures have advanced beyond earlier models by developing robust structural encodings and enabling scalability for medium-to-large graphs. To enhance the structural expressiveness of node tokens, various positional and structural encoding techniques have been introduced Dwivedi et al. (2022a); Cantürk et al. (2024); Lim et al. (2022); Huang et al. (2024); Kanatsoulis et al. (2025), effectively embedding the input graph’s topology into the model.

Spectral methods in GNNs leverage the graph Laplacian’s eigenstructure to encode node positions and capture topological properties, addressing limitations in traditional message-passing GNNs like over-squashing and limited expressiveness. Early work, such as ChebNet Defferrard et al. (2016), introduced spectral convolutions by approximating filters on the Laplacian spectrum. Laplacian Positional Encoding (LapPE), as utilized in GraphGPS Rampásek et al. (2022), employs the smallest non-zero eigenvalues and eigenvectors of the normalized Laplacian to provide node-specific positional signals, enhancing structural awareness in graph transformers.

Recent advancements have focused on improving the spectral encodings’ uniqueness. The sheaf Laplacian perturbation method Choi et al. (2025) introduces controlled perturbations to the sheaf Laplacian, increasing eigenvalue diversity and expressiveness to distinguish isomorphic substructures, surpassing 1-WL limitations. Spatio-Spectral GNNs (S^2 GNNs) Geisler et al. (2024) combine spatial message-passing with spectral filtering, mitigating over-squashing and achieving higher-order WL expressiveness through optimized filter banks. For dynamic graphs, Supra-Laplacian Encoding (SLATE) Karmim et al. (2024) constructs supra-Laplacians to capture temporal dynamics, enhancing spatio-temporal transformers for tasks like forecasting. Comprehensive surveys, such as Shehzad et al. (2024), categorize graph transformer architectures, highlighting spectral encodings’ role in improving permutation equivariance and expressivity.

3 GRAPHFRACTALNET ARCHITECTURE

In this section, we present GraphFractalNet, a novel Transformer-based architecture designed to be both expressive and scalable for graph representation learning. Rather than relying on handcrafted structural encodings, GraphFractalNet leverages spectral embeddings derived from the graph Laplacian, enabling it to capture both local and global structural properties in a principled and data-driven way. To improve scalability and adaptability, we introduce a dynamic graph rewiring mechanism that modifies the graph topology layer-wise based on learned relevance scores, reducing computational overhead while preserving critical connectivity. Moreover, we design a fractal-inspired attention mechanism that enforces sparse, hierarchical interactions by restricting attention computation to self-similar patterns identified via recursive graph partitioning. This leads to sublinear complexity in practice, without sacrificing expressiveness. Altogether, these components allow GraphFractalNet to generalize beyond the limitations of existing graph Transformers, enabling efficient and effective representation learning on large and structurally complex graphs.

3.1 PRELIMINARIES

Let $G = (V, E)$ denote a graph with $|V| = N$ nodes and edges E . Each node $v_i \in V$ has a feature vector $x_i \in \mathbb{R}^d$, and each edge $e_{ij} \in E$ has a feature vector $x_e^{ij} \in \mathbb{R}^{d_e}$. The adjacency matrix is $A \in \{0, 1\}^{N \times N}$, and the degree matrix is D . The normalized graph Laplacian is $L = I - D^{-1/2} A D^{-1/2}$. The goal is to learn a graph representation $h_G \in \mathbb{R}^d$ or node representations $\{h_i\}_{i=1}^N$.

3.2 SPECTRAL ENCODER

A key challenge in graph representation learning is capturing both global topology and local connectivity. Traditional positional encodings like shortest-path distances or centrality are limited and inflexible across diverse graphs. To address this, GraphFractalNet introduces a spectral encoder that maps nodes and edges into a spectral domain derived from the graph Laplacian, enabling principled, learnable, and multi-scale representations.

The normalized Laplacian L admits an eigendecomposition $L = \Phi \Lambda \Phi^T$, where $\Phi \in \mathbb{R}^{N \times N}$ contains orthonormal eigenvectors and Λ is a diagonal matrix of eigenvalues. These spectral components capture the intrinsic geometry of the graph and underpin our encoding framework.

Given node features $X \in \mathbb{R}^{N \times d}$, we compute spectral embeddings as:

$$X_s = \Phi^T X W_s, \quad (1)$$

where $W_s \in \mathbb{R}^{d \times d_s}$ is a learnable matrix. Low-frequency components encode global structure, while high-frequency components capture local detail, enabling multi-scale reasoning.

To enrich edge representations, we define spectral edge embeddings as:

$$x_{s,e}^{ij} = x_e^{ij} W_{s,e} + (\phi_i - \phi_j)^T W_{s,e}^\phi, \quad (2)$$

where x_e^{ij} is the edge feature, and $W_{s,e}$, $W_{s,e}^\phi$ are learnable parameters. The spectral difference $(\phi_i - \phi_j)$ encodes structural contrast between connected nodes.

This spectral encoder provides a unified, geometry-aware representation space that supports both local and global reasoning. Unlike handcrafted encodings, spectral embeddings are continuous, learnable, and stable across layers, enabling effective downstream processing by modules like fractal attention and dynamic rewiring.

3.3 DYNAMIC GRAPH REWIRING MODULE

To enhance computational efficiency and adaptively refine the graph structure, GraphFractalNet employs a dynamic graph rewiring mechanism that reconstructs the graph topology at each layer based on learned relevance scores. For each node pair (i, j) , a connectivity relevance score is computed as:

$$R_{ij}^{(l)} = \sigma \left(\frac{(h_i^{(l-1)} W_r) \cdot (h_j^{(l-1)} W_r)}{\sqrt{d_r}} + \beta \cdot \text{SPD}(i, j) \right), \quad (3)$$

where $h_i^{(l-1)} \in \mathbb{R}^d$ is the node representation from the previous layer, $W_r \in \mathbb{R}^{d \times d_r}$ is a learnable projection matrix, σ denotes the sigmoid activation, $\text{SPD}(i, j)$ is the shortest-path distance between nodes i and j , and $\beta \in \mathbb{R}$ is a learnable scaling factor.

Based on these scores, we select the top $k \log N$ most relevant edges per node to construct a new, sparse adjacency matrix $A^{(l)}$, effectively rewiring the graph $G^{(l)} = (V, E^{(l)})$ for layer l . To maintain essential structural information and prevent graph disconnection, we introduce a probabilistic edge retention mechanism over original edges:

$$P(\text{keep } e_{ij}) = \min \left(1, \frac{\gamma}{1 + \text{SPD}(i, j)} \right), \quad (4)$$

where γ is a tunable hyperparameter. This strategy favors the retention of shorter, more informative edges while allowing flexibility in pruning redundant connections.

The resulting rewired graph is both sparser and more expressive, enabling the model to dynamically adapt its receptive field, focus computation on structurally significant regions, and scale to large graphs without compromising representational fidelity.

3.4 FRACTAL ATTENTION LAYER

We introduce a Fractal Attention Mechanism, inspired by fractal geometry, to effectively capture the hierarchical and self-similar structures that characterize graphs. This mechanism leverages recursive graph partitioning to constrain attention within local and multi-scale neighborhoods, enabling scalable and structured information flow.

At each layer l , we perform a hierarchical spectral clustering that partitions the graph into k_l clusters, forming a tree-like fractal hierarchy of subgraphs. Based on this hierarchy, we construct a fractal attention mask $M_f \in \mathbb{R}^{N \times N}$, which restricts attention to node pairs that belong to the same or neighboring clusters at the current recursion level.

Attention scores are computed using the spectral node representations $H_s^{(l)} \in \mathbb{R}^{N \times d_s}$ as follows:

$$A_f = \text{softmax} \left(\frac{(H_s^{(l)} W_{Q,f})(H_s^{(l)} W_{K,f})^T}{\sqrt{d_f}} \odot M_f \right), \quad (5)$$

where $W_{Q,f}, W_{K,f} \in \mathbb{R}^{d_s \times d_f}$ are learnable projection matrices, and $\odot$ denotes element-wise multiplication. The mask $M_f^{(l)}$ is defined recursively:

$$M_f^{(l)}(i, j) = \begin{cases} 1 & \text{if } i, j \text{ in same or adjacent clusters at level } l \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The final output of the fractal attention layer is computed as:

$$H_f^{(l)} = A_f(H_s^{(l)} W_{V,f}), \quad (7)$$

where $W_{V,f} \in \mathbb{R}^{d_s \times d}$ is a learnable value projection.

By restricting attention within a recursively structured hierarchy, the fractal attention mechanism significantly reduces computational overhead, achieving a complexity of $\mathcal{O}(N \log \log N)$. This design enables efficient modeling of both local and global dependencies, while maintaining scalability for large graphs.

3.5 SPECTRAL MESSAGE PASSING

We propose a spectral message passing mechanism that integrates edge attributes with spectral embeddings, operating over the dynamically rewired graph structure. At each layer l , the message for a node i is computed by aggregating information from its neighbors in the rewired graph $\mathcal{N}^{(l)}(i)$:

$$m_i^{(l)} = \sum_{j \in \mathcal{N}^{(l)}(i)} A_{ij}^{(l)} \cdot \langle x_{s,e}^{ij}, W_m x_{s,j} \rangle, \quad (8)$$

where $x_{s,j} \in \mathbb{R}^{d_s}$ is the spectral embedding of node j , $x_{s,e}^{ij}$ is the spectral edge embedding between nodes i and j , $W_m \in \mathbb{R}^{d_s \times d_s}$ is a learnable transformation matrix, and $\langle \cdot, \cdot \rangle$ denotes the dot product in the spectral space. The attention weight $A_{ij}^{(l)}$ modulates the influence of neighbor j on node i at layer l .

The updated node representation is then obtained via a residual connection followed by layer normalization:

$$h_i^{(l)} = \text{LN} \left(h_{f,i}^{(l)} + m_i^{(l)} \right), \quad (9)$$

where $h_{f,i}^{(l)}$ is the output of the fractal attention layer. This formulation allows the model to dynamically propagate information using both node and edge-level spectral features, while preserving the stability and expressivity of message passing in the spectral domain.

3.6 HIERARCHICAL READOUT MODULE

To obtain a comprehensive graph-level representation, we introduce a hierarchical readout mechanism that aggregates multi-scale features across layers. Specifically, a virtual node v_0 is added to the graph and connected to all other nodes. This virtual node acts as a global aggregator and is updated at each layer using the fractal attention mechanism, allowing it to capture both local and global context.

The final graph representation is computed by aggregating the virtual node’s representations across all layers:

$$h_G = \text{MLP} \left(\sum_{l=1}^L \alpha_l h_0^{(l)} \right), \quad (10)$$

where $h_0^{(l)}$ denotes the virtual node’s embedding at layer l , α_l are learnable layer-wise importance weights, and MLP denotes a multi-layer perceptron that refines the aggregated representation. This hierarchical readout enables the model to adaptively combine structural information captured at different depths of the network for robust graph-level prediction.

3.7 GRAPHFRACTALNET LAYER

Each layer of GraphFractalNet unifies spectral encoding, fractal attention, and message passing within a structured computational block designed to capture both global and local graph patterns. Given the node representations from the previous layer, denoted by $H^{(l-1)}$, the layer first performs dynamic graph rewiring, adapting the graph topology based on feature similarity. This step produces a new adjacency matrix $A^{(l)}$, allowing the network to refine the neighborhood structure at each layer and better capture task-relevant dependencies.

To encode global structural information, the node features are projected into the spectral domain using the top- k eigenvectors Φ of the normalized Laplacian. This yields the spectral representation $H_s^{(l)} = \Phi^\top H^{(l-1)} W_s$, where W_s is a learnable weight matrix. The spectral features are then passed through the fractal attention module, which leverages a recursively designed attention mask to capture multi-scale dependencies. The resulting attended representation, denoted $H_f^{(l)}$, captures hierarchical structure while preserving spatial and spectral coherence.

Next, the model performs spectral message passing, integrating edge features and neighborhood information from the rewired graph. Using the formulation described earlier, messages are aggregated in the spectral space and added to the fractal-attended node features. This yields an updated node representation, which is then normalized and passed through a feed-forward network (FFN), following a residual connection. The final output of the layer is given by:

$$H^{(l)} = \text{FFN} \left(\text{LN} \left(H^{(l-1)} + \mathcal{M}_{\text{spec}}(H_f^{(l)}, A^{(l)}, X_{s,e}) \right) \right) + H^{(l-1)}, \quad (11)$$

where $\mathcal{M}_{\text{spec}}(\cdot)$ denotes the spectral message passing operation introduced in the previous subsection. It integrates edge-aware interactions in the spectral domain, leveraging the rewired adjacency matrix $A^{(l)}$, the fractal-attended spectral features $H_f^{(l)}$, and the spectral edge embeddings $X_{s,e}$. This operation allows the model to capture structurally informed messages across dynamically evolving neighborhoods, enabling robust propagation of both global and local information through the graph.

This composition allows each layer to progressively refine node representations by jointly modeling local connectivity, spectral structure, and self-similarity across the graph hierarchy. Appendix A analyzes the computational efficiency of GraphFractalNet, while Appendix B establishes its expressiveness with formal theorems. Appendix C further examines its power in modeling long-range dependencies and structural patterns.

4 EXPERIMENTS

Datasets To thoroughly evaluate the effectiveness of GraphFractalNet, we assess its performance across a diverse set of graph learning tasks, as described below and in Appendix D.

- **Graph-Level Prediction.** For graph classification and regression, we use five standard benchmarks from Benchmarking GNNs Dwivedi et al. (2023a): ZINC, MNIST, CIFAR10, PATTERN, and CLUSTER. In addition, we include two long-range molecular benchmarks Dwivedi et al. (2022b) from the LRGB suite: Peptides-func (10-class classification) and Peptides-struct (multi-property regression), following the standard evaluation protocols Rampásek et al. (2022).
- **Node Classification.** We consider a range of node classification benchmarks covering both medium- and large-scale graphs. The medium-scale datasets include citation networks Kipf (2017) (Cora, CiteSeer, PubMed), an actor co-occurrence graph Chien et al. (2021), and heterophilic page-page networks such as Squirrel and Chameleon Rozemberczki et al. (2021). To further test scalability, we adopt large-scale datasets from the Open Graph Benchmark (OGB) Hu et al. (2020), including ogbn-arxiv, arxiv-year, ogbn-proteins, ogbn-products, and the extremely large ogbn-papers100M, with graph sizes ranging from thousands to over 100 million nodes.

Baseline Models for Comparison To rigorously evaluate the effectiveness of GraphFractalNet, we benchmark it against a broad spectrum of strong baselines, spanning both message-passing GNNs and transformer architectures.

- **Classical and Modern GNNs.** We include widely used message-passing models such as GCN Kipf (2017), GIN Xu et al. (2019), and GAT Casanova et al. (2018), along with enhanced variants like GatedGCN Bresson & Laurent (2017), GatedGCN-RWSE Dwivedi et al. (2022a), and PNA Corso et al. (2020). We further consider more advanced designs tailored for challenging benchmarks, including SIGN Frasca et al. (2020), LINKX Lim et al. (2021), CIN Bodnar et al. (2021), GIN-AK+ Zhao et al. (2022), and HC-GNN Zhong et al. (2023), many of which represent the latest state-of-the-art in graph neural networks.
- **Graph Transformers.** To capture the comparison against attention-based and structural encoding methods, we evaluate a rich set of graph transformer models. These include GT Dwivedi & Bresson (2020), Graphormer Ying et al. (2021), SAN Kreuzer et al. (2021), ANS-GT Zhang et al. (2022), EGT Hussain et al. (2022), NodeFormer Wu et al. (2022), SpecFormer Bo et al. (2023), MGT Ngo et al. (2023), AGT Ma et al. (2023b), HSGT Zhu et al. (2023), Graphormer-GD Zhang et al. (2023), SAT Chen et al. (2022), GOAT Kong et al. (2023), GapFormer Liu et al. (2023), Graph ViT/MLP-Mixer He et al. (2023), LargeGT Dwivedi et al. (2023b), NAGphormer Chen et al. (2023), CoBFormer Xing et al. (2024), Expformer Shirzad et al. (2023), DRew Gutteridge et al. (2023), and VCR-GT Fu et al. (2024). Finally, we benchmark against the latest graph transformer advancements such as GraphGPS Rampásek et al. (2022), GRIT Ma et al. (2023a), and SGFormer Wu et al. (2023), which represent cutting-edge approaches in scalability, structural encoding, and generalization.

This wide coverage ensures that the evaluation of GraphFractalNet reflects performance against both long-established baselines and the most recent state-of-the-art models.

Experimental Setup. Our hyperparameters are selected within the grid search space defined by SGFormer. All other experimental settings, such as dropout, batch size, training strategy, and optimizer, follow the same configuration as SGFormer Wu et al. (2023). For evaluation, we report the test accuracy of the model that achieves the best performance on the validation set. Each experiment is conducted 10 times, and we present the mean accuracy together with the corresponding error bars.

4.1 DISCUSSION OF GRAPH-LEVEL RESULTS

The results in Table 1 demonstrate that GraphFractalNet consistently outperforms a wide range of competitive baselines across multiple graph-level benchmarks. On ZINC, our model achieves the lowest MAE (0.052), surpassing recent spectral and transformer-based methods such as Specformer (0.066) and GRIT (0.059). For classification tasks, GraphFractalNet attains the highest accuracy on MNIST (98.69), CIFAR10 (76.89), and PATTERN (87.68), while maintaining competitive performance on CLUSTER. Notably, the improvements are most pronounced for baselines like GT, where the integration of spectral encoding and fractal attention significantly enhances expressivity, narrowing the gap with deeper or more specialized models. Although GRIT remains very strong

due to its distance-aware random walk probabilities (RRWP), GraphFractalNet still secures superior or second-best performance on most datasets, indicating that our design complements existing structural encodings.

Turning to the Long-Range Graph Benchmark results in Table 2, GraphFractalNet again demonstrates substantial gains. On Peptides-func, it achieves an AP of 0.7289, clearly outperforming all other baselines, including DRew (0.7150) and GRIT (0.6988). This 6.21% improvement highlights the model’s strength in capturing functional diversity in molecular graphs. On Peptides-struct, GraphFractalNet attains an MAE of 0.2462, which is competitive with state-of-the-art models such as MGT+WavePE (0.2453) and Exphormer (0.2481), while still surpassing several strong baselines.

Overall, these results provide strong evidence that GraphFractalNet’s combination of spectral embeddings, dynamic rewiring, and fractal attention yields a model that is both scalable and highly expressive, capable of matching or exceeding the performance of specialized spectral or attention-based graph transformers across diverse tasks.

Table 1: The results are presented as the mean $\pm$ standard deviation from 5 runs using different random seeds. Highlighted are the top **first**, **second** and **third** results.

Model	ZINC MAE $\downarrow$	MNIST Accuracy $\uparrow$	CIFAR10 Accuracy $\uparrow$	PATTERN Accuracy $\uparrow$	CLUSTER Accuracy $\uparrow$
GCN	0.367 $\pm$ 0.011	90.705 $\pm$ 0.218	55.710 $\pm$ 0.381	71.892 $\pm$ 0.334	68.498 $\pm$ 0.976
GatedGCN	0.282 $\pm$ 0.015	97.340 $\pm$ 0.143	67.312 $\pm$ 0.311	85.568 $\pm$ 0.088	73.840 $\pm$ 0.326
GIN-AK+	0.080 $\pm$ 0.001	—	72.190 $\pm$ 0.130	86.850 $\pm$ 0.057	—
GIN	0.526 $\pm$ 0.051	96.485 $\pm$ 0.252	55.255 $\pm$ 1.527	85.387 $\pm$ 0.136	64.716 $\pm$ 1.553
CIN	0.079 $\pm$ 0.006	—	—	—	—
PNA	0.188 $\pm$ 0.004	97.940 $\pm$ 0.120	70.350 $\pm$ 0.630	—	—
SAN	0.139 $\pm$ 0.006	—	—	86.581 $\pm$ 0.037	76.691 $\pm$ 0.650
Graphormer-GD	0.081 $\pm$ 0.009	—	—	—	—
SGFormer	0.306 $\pm$ 0.023	—	—	85.287 $\pm$ 0.097	69.972 $\pm$ 0.634
EGT	0.108 $\pm$ 0.009	98.173 $\pm$ 0.087	68.702 $\pm$ 0.409	86.821 $\pm$ 0.020	79.232 $\pm$ 0.348
Specformer	0.066 $\pm$ 0.003	—	—	—	—
Exphormer	—	98.550 $\pm$ 0.039	74.696 $\pm$ 0.125	86.742 $\pm$ 0.015	78.071 $\pm$ 0.037
SAT	0.094 $\pm$ 0.008	—	—	86.848 $\pm$ 0.037	77.856 $\pm$ 0.104
Graph ViT/MLP-Mixer	0.073 $\pm$ 0.001	97.422 $\pm$ 0.110	73.961 $\pm$ 0.330	—	—
GraphGPS	0.070 $\pm$ 0.004	98.051 $\pm$ 0.126	72.298 $\pm$ 0.356	86.685 $\pm$ 0.059	78.016 $\pm$ 0.180
GRIT	0.059 $\pm$ 0.002	98.108 $\pm$ 0.111	76.468 $\pm$ 0.881	87.196 $\pm$ 0.076	80.026 $\pm$ 0.277
GT	0.226 $\pm$ 0.014	90.831 $\pm$ 0.161	59.753 $\pm$ 0.293	84.808 $\pm$ 0.068	73.169 $\pm$ 0.622
GraphFractalNet	0.052 $\pm$ 0.008	98.692 $\pm$ 0.362	76.892 $\pm$ 0.362	87.684 $\pm$ 0.218	79.267 $\pm$ 0.196

Table 2: Test performance on two peptide datasets from Long-Range Graph Benchmarks (LRGB).

Model	Peptides-func AP $\uparrow$	Peptides-struct MAE $\downarrow$
GCN	0.5930 $\pm$ 0.0023	0.3496 $\pm$ 0.0013
GatedGCN+RWSE	0.6069 $\pm$ 0.0035	0.3357 $\pm$ 0.0006
GatedGCN	0.5864 $\pm$ 0.0035	0.3420 $\pm$ 0.0013
SAN+RWSE	0.6439 $\pm$ 0.0075	0.2545 $\pm$ 0.0012
GT	0.6326 $\pm$ 0.0126	0.2529 $\pm$ 0.0016
MGT+WavePE	0.6817 $\pm$ 0.0064	0.2453 $\pm$ 0.0025
GRIT	0.6988 $\pm$ 0.0082	0.2460 $\pm$ 0.0012
DRew	0.7150 $\pm$ 0.0044	0.2536 $\pm$ 0.0015
Graph ViT/MLP-Mixer	0.6970 $\pm$ 0.0080	0.2475 $\pm$ 0.0015
Exphormer	0.6527 $\pm$ 0.0043	0.2481 $\pm$ 0.0007
GraphGPS	0.6535 $\pm$ 0.0041	0.2500 $\pm$ 0.0012
GraphFractalNet	0.7289 $\pm$ 0.0397	0.2462 $\pm$ 0.0082

4.2 RESULTS ON LARGE-SCALE GRAPHS

Overall Performance. The results in Table 3 highlight the competitiveness of GraphFractalNet across a wide spectrum of benchmarks, ranging from citation networks to extremely large-scale datasets such as ogbn-products and ogbn-papers100M. On small and medium-sized datasets (e.g., Cora, CiteSeer, PubMed), GraphFractalNet consistently achieves superior accuracy compared to both traditional GNNs (HC-GNN, SIGN, LINKX) and recent graph transformers (Graphormer, SAT). Notably, it achieves state-of-the-art performance on challenging heterophilic datasets such as Squirrel and Chameleon, where many existing methods tend to underperform due to oversmoothing or lack of structural bias. When scaling to large graphs (ogbn-proteins, ogbn-arxiv, ogbn-products, ogbn-100M), GraphFractalNet maintains strong performance while avoiding the memory bottlenecks observed in several transformer-based baselines. Models such as Graphormer, SAT, and Exphormer frequently encounter out-of-memory (OOM) issues even on GPUs with 40GB memory,

Table 3: Node classification on large-scale graphs (%). OOM indicates out-of-memory when training on a GPU with 40GB of memory.

Model	Cora	CiteSeer	PubMed	Actor	Squirrel	Chameleon	ogbn-proteins	ogbn-arxiv	arxiv-year	ogbn-products	ogbn-papers100M
# edges	5,278	4,552	44,324	29,926	46,998	8,854	39,561,252	1,166,243	1,166,243	61,859,140	1,615,685,872
# nodes	2,708	3,327	19,717	7,600	2,223	890	132,534	169,343	169,343	2,449,029	111,059,956
	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	ROC-AUC↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑
HC-GNN	81.9 ± 0.4	72.5 ± 0.6	80.2 ± 0.6	-	-	-	-	72.79 ± 0.25	-	-	-
LINKX	-	72.5	-	36.1 ± 1.5	41.9 ± 1.2	43.8 ± 2.9	66.18 ± 0.33	53.53 ± 0.36	71.59 ± 0.71	-	-
SIGN	82.1 ± 0.3	72.4 ± 0.8	79.5 ± 0.5	36.5 ± 1.0	40.7 ± 2.5	41.7 ± 2.2	71.24 ± 0.46	71.95 ± 0.11	-	80.52 ± 0.16	65.11 ± 0.14
AGT	81.7 ± 0.4	71.0 ± 0.6	-	-	-	-	-	72.28 ± 0.38	47.38 ± 0.78	-	-
HSGT	83.6 ± 1.8	67.4 ± 0.9	79.7 ± 0.5	-	-	-	78.13 ± 0.25	72.58 ± 0.31	-	81.15 ± 0.13	-
ANS-GT	79.4 ± 0.9	64.5 ± 0.7	77.8 ± 0.7	35.2 ± 1.3	40.8 ± 2.1	42.6 ± 2.7	74.67 ± 0.65	72.34 ± 0.50	-	80.64 ± 0.29	-
Graphormer	75.8 ± 1.1	65.6 ± 0.6	OOM	OOM	40.9 ± 2.5	41.9 ± 2.8	OOM	OOM	OOM	OOM	OOM
SAT	72.4 ± 0.3	60.9 ± 1.3	OOM	-	-	-	OOM	OOM	OOM	OOM	OOM
Gapformer	83.5 ± 0.4	71.4 ± 0.6	80.2 ± 0.4	-	-	-	-	71.90 ± 0.19	-	-	-
GraphGPS	76.5 ± 0.6	-	65.7 ± 1.0	33.1 ± 0.8	-	36.2 ± 0.6	-	70.97 ± 0.41	-	OOM	OOM
NAGphormer	-	-	-	34.3 ± 0.9	39.7 ± 0.7	40.3 ± 1.7	-	70.13 ± 0.55	-	73.55 ± 0.21	-
LargeGT	-	-	-	-	-	-	-	-	-	-	64.73 ± 0.05
VCR-GT	-	-	-	-	-	-	-	-	54.15 ± 0.09	-	-
SGFormer	84.5 ± 0.8	72.6 ± 0.2	80.3 ± 0.6	37.9 ± 1.1	41.8 ± 2.2	44.9 ± 3.9	79.53 ± 0.38	72.63 ± 0.33	-	75.36 ± 0.33	66.01 ± 0.37
NodeFormer	82.2 ± 0.9	72.5 ± 1.1	79.9 ± 1.0	36.9 ± 1.0	38.5 ± 1.5	34.7 ± 4.1	77.45 ± 1.15	59.90 ± 0.42	-	72.93 ± 0.13	-
CoBFormer	-	-	-	37.4 ± 1.0	-	-	-	73.17 ± 0.18	-	78.15 ± 0.07	-
Expformer	-	-	-	-	-	-	-	72.44 ± 0.28	-	OOM	OOM
GOAT	82.1 ± 0.9	71.6 ± 1.3	78.9 ± 1.2	32.1 ± 1.8	41.1 ± 2.5	43.5 ± 2.3	78.37 ± 0.26	72.41 ± 0.40	53.57 ± 0.18	82.00 ± 0.43	65.05 ± 0.13
GraphFractalNet	84.87 ± 0.6	73.76 ± 0.4	80.46 ± 1.0	37.6 ± 1.3	43.87 ± 1.6	46.74 ± 1.8	81.26 ± 0.82	72.94 ± 0.42	54.73 ± 0.18	82.47 ± 0.42	65.89 ± 0.16

Table 4: Efficiency comparison of GraphFractalNet and scalable graph transformer competitors; training time per epoch.

	PubMed	ogbn-proteins	ogbn-arxiv	ogbn-products	ogbn-papers100M
SGFormer	15.4ms	0.8s	0.2s	4.8s	579.4s
NodeFormer	321.4ms	1.8s	0.6s	5.6s	595.1s
GraphFractalNet	14.4ms	0.5s	0.3s	4.7s	474.5s

whereas GraphFractalNet preserves efficiency without compromising accuracy. Importantly, the integration of fractal-inspired hierarchical encoding allows it to capture global structural information while retaining discriminative power for node-level tasks, leading to a balanced improvement in both homophilic and heterophilic settings.

Efficiency Comparison. Table 4 further emphasizes the scalability advantage of GraphFractalNet. The training time per epoch is consistently lower or competitive with the fastest existing scalable graph transformers. For instance, on PubMed and ogbn-proteins, GraphFractalNet is faster than SGFormer while simultaneously delivering higher predictive accuracy (as shown in Table 3). Even on extremely large graphs like ogbn-papers100M, GraphFractalNet reduces training time by more than 100 seconds per epoch compared to NodeFormer, demonstrating its ability to handle billion-scale edges without significant overhead. This balance between accuracy and efficiency underscores the model’s practicality for real-world large-scale applications.

We also conduct experiments on large-scale heterophilic networks, with results presented in Appendix E.1. Additionally, we perform a sensitivity analysis (Appendix E.2 and E.3) and an ablation study (Appendix E.4).

5 CONCLUSIONS

In this paper, we proposed GraphFractalNet, a novel graph transformer that incorporates fractal-inspired hierarchical encoding, spectral positional features, and dynamic rewiring to capture both local and global structural dependencies in graphs. Our extensive experiments demonstrate that GraphFractalNet consistently achieves superior or competitive performance across diverse benchmarks, including homophilic, heterophilic, and large-scale datasets, while maintaining efficiency. Unlike several existing graph transformers that often face scalability challenges or out-of-memory issues, our model provides robust accuracy with reduced computational cost and faster training speed. The results further highlight its ability to generalize across different graph structures, from small citation networks to billion-scale graphs. These advantages position GraphFractalNet as a strong and scalable backbone for graph learning tasks.

REFERENCES

- Louis Airale, Antonio Longa, Mattia Rigon, Andrea Passerini, and Roberto Passerone. Simple path structural encoding for graph transformers. *Forty-Second International Conference on Machine Learning*, 2025.
- Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *International Conference on Learning Representations*, 2021.
- Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
- Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. *International Conference on Learning Representations*, 2023.
- Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.
- Tomer Borreda, Daniel Freedman, and Or Litany. ReHub: Linear complexity graph transformers with adaptive hub-spoke reassignment. *arXiv preprint arXiv:2412.01519*, 2024.
- Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.
- Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. *Proceedings of the 41st International Conference on Machine Learning*, 2024.
- PVGCA Casanova, Adriana Romero Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR. Petar Velickovic Guillem Cucurull Arantxa Casanova Adriana Romero Pietro Liò and Yoshua Bengio*, 2018.
- Dexiong Chen, Leslie O’Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pp. 3469–3489. PMLR, 2022.
- Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. *International Conference on Learning Representations*, 2023.
- Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *International Conference on Learning Representations*, 2021.
- Yoonhyuk Choi, Jiho Choi, and Chong-Kwon Kim. Sheaf graph neural networks via pac-bayes spectral optimization. *arXiv preprint arXiv:2508.00357*, 2025.
- Yun Young Choi, Sun Woo Park, Minhoo Lee, and Youngho Woo. Topology-informed graph transformer. *41st International Conference on Machine Learning*, 2024.
- Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271, 2020.
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.
- Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

- Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *International Conference on Learning Representations*, 2022a.
- Vijay Prakash Dwivedi, Ladislav Rampásek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022b.
- Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023a.
- Vijay Prakash Dwivedi, Yozen Liu, Anh Tuan Luu, Xavier Bresson, Neil Shah, and Tong Zhao. Graph transformers for large graphs. *arXiv preprint arXiv:2312.11109*, 2023b.
- Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.
- Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. Vcr-graphormer: A mini-batch graph transformer via virtual connections. *International Conference on Learning Representations*, 2024.
- Simon Markus Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. Spatio-spectral graph neural networks. *Advances in Neural Information Processing Systems*, 37:49022–49080, 2024.
- Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pp. 12252–12267. PMLR, 2023.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. In *International Conference on Machine Learning*, pp. 12724–12745. PMLR, 2023.
- Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- Yinan Huang, William Lu, Joshua Robinson, Yu Yang, Muhan Zhang, Stefanie Jegelka, and Pan Li. On the stability of expressive positional encodings for graphs. *International Conference on Learning Representations*, 2024.
- Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Global self-attention as a replacement for graph convolution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 655–665, 2022.
- Chaitanya K Joshi. Transformers are graph neural networks. *arXiv preprint arXiv:2506.22084*, 2025.
- Charilaos I Kanatsoulis, Evelyn Choi, Stephanie Jegelka, Jure Leskovec, and Alejandro Ribeiro. Learning efficient positional encodings with graph neural networks. *arXiv preprint arXiv:2502.01122*, 2025.
- Yannis Karmim, Marc Lafon, Raphaël Fournier-S’Niehotta, and Nicolas Thome. Supra-laplacian encoding for transformer on dynamic graphs. *Advances in Neural Information Processing Systems*, 37:17215–17246, 2024.
- T N Kipf. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

- Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein. GOAT: A global transformer on large-scale graphs. In *International Conference on Machine Learning*, pp. 17375–17390. PMLR, 2023.
- Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.
- Zhonghao Li, Ji Shi, Xinming Zhang, Miao Zhang, and Bo Li. SFi-Former: Sparse flow induced attention for graph transformer. In *Proceedings of the 2025 International Conference on Multi-media Retrieval*, pp. 760–769, 2025.
- Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.
- Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.
- Chuang Liu, Yibing Zhan, Xueqi Ma, Liang Ding, Dapeng Tao, Jia Wu, and Wenbin Hu. Gapformer: Graph transformer with graph pooling for node classification. In *IJCAI*, pp. 2196–2205, 2023.
- Yuankai Luo, Hongkang Li, Lei Shi, and Xiao-Ming Wu. Enhancing graph transformers with hierarchical distance structural encoding. *Advances in Neural Information Processing Systems*, 37: 57150–57182, 2024.
- Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, pp. 23321–23337. PMLR, 2023a.
- Xiaojun Ma, Qin Chen, Yi Wu, Guojie Song, Liang Wang, and Bo Zheng. Rethinking structural encodings: Adaptive graph transformer for node classification task. In *Proceedings of the ACM Web Conference 2023*, pp. 533–544, 2023b.
- Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.
- Nhat Khang Ngo, Truong Son Hy, and Risi Kondor. Multiresolution graph transformers and wavelet positional encoding for learning long-range and hierarchical structures. *The Journal of Chemical Physics*, 159(3), 2023.
- Taoyang Qin, CHEN Ke-Jia, and Zheng Liu. Enhancing spectral GNNs: From topology and perturbation perspectives. In *Forty-second International Conference on Machine Learning*, 2025.
- Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.
- Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.
- Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *arXiv preprint arXiv:2407.09777*, 2024.
- Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pp. 31613–31632. PMLR, 2023.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401, 2022.
- Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. SGformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.
- Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. Less is more: on the over-globalizing problem in graph transformers. *Proceedings of the 41st International Conference on Machine Learning*, 2024.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations*, 2019.
- Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.
- Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of GNNs via graph biconnectivity. *arXiv preprint arXiv:2301.09505*, 2023.
- Guibin Zhang, Yanwei Yue, Kun Wang, Junfeng Fang, Yongduo Sui, Kai Wang, Yuxuan Liang, Dawei Cheng, Shirui Pan, and Tianlong Chen. Two heads are better than one: Boosting graph sparse training via semantic and topological awareness. *arXiv preprint arXiv:2402.01242*, 2024.
- Zaixi Zhang, Qi Liu, Qingyong Hu, and Chee-Kong Lee. Hierarchical graph transformer with adaptive node sampling. *Advances in Neural Information Processing Systems*, 35:21171–21183, 2022.
- Haiteng Zhao, Shuming Ma, Dongdong Zhang, Zhi-Hong Deng, and Furu Wei. Are more layers beneficial to graph transformers? *International Conference on Learning Representations*, 2023.
- Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any gnn with local structure awareness. *International Conference on Learning Representations*, 2022.
- Zhiqiang Zhong, Cheng-Te Li, and Jun Pang. Hierarchical message-passing graph neural networks. *Data Mining and Knowledge Discovery*, 37(1):381–408, 2023.
- Wenhao Zhu, Tianyu Wen, Guojie Song, Xiaojun Ma, and Liang Wang. Hierarchical transformer for scalable graph learning. *arXiv preprint arXiv:2305.02866*, 2023.

A COMPLEXITY ANALYSIS

GraphFractalNet is designed to achieve both expressive power and computational efficiency. The overall complexity of the model per layer is significantly lower than that of traditional transformer-based graph models. The spectral embedding step, computed via an approximate eigen decomposition (e.g., the Lanczos method), incurs a cost of $O(N \log N)$, where N is the number of nodes. The dynamic graph rewiring procedure, which selects structurally relevant edges based on learned relevance scores, also operates in $O(N \log N)$. Fractal attention further enhances scalability through hierarchical clustering and sparse attention computation, resulting in a sub-logarithmic cost of $O(N \log \log N)$. The spectral message-passing operation, applied over the sparsified, rewired graph $E^{(l)}$, scales linearly with the number of edges, i.e., $O(|E^{(l)}|)$, which is upper bounded by $O(N \log N)$. Consequently, the overall per-layer complexity of GraphFractalNet is $O(N \log \log N)$, representing a substantial reduction compared to the quadratic complexity $O(N^2)$ observed in standard transformer-based graph models. This improvement allows GraphFractalNet to scale more efficiently to large graphs while preserving its ability to capture multi-scale structural dependencies.

B EXPRESSIVENESS

GraphFractalNet exhibits strong expressive power, going beyond traditional message-passing GNNs and the Weisfeiler-Lehman (WL) hierarchy. Specifically, GraphFractalNet can simulate common GNN architectures such as GCN and GIN through suitable choices of spectral bases and attention configurations. This is due to the modular design of our network, where the spectral embedding step encodes both global and local topological signals, while the fractal attention mechanism adaptively focuses on structurally meaningful subgraphs across multiple scales.

The incorporation of spectral embeddings allows GraphFractalNet to capture global structural signatures via the eigenbasis of the graph Laplacian, while the fractal attention introduces hierarchical context aggregation that is sensitive to non-local interactions and motif-level patterns. Together, these components empower GraphFractalNet to distinguish graphs that cannot be resolved by the k -WL test for $k \geq 2$, by going beyond neighborhood isomorphism and capturing richer topological invariants.

Theorem 1. *GraphFractalNet is strictly more expressive than the k -Weisfeiler–Lehman (k -WL) test for $k \geq 2$.*

Proof. Let $G = (V, E)$ and $G' = (V', E')$ be two non-isomorphic graphs such that:

$$\text{k-WL}(G) = \text{k-WL}(G') \quad \text{for some } k \geq 2, \quad (12)$$

i.e., the k -WL test fails to distinguish G and G' .

Assume that both graphs are regular and share identical local k -tuple neighborhoods. However, let their Laplacian spectra differ:

$$\text{spec}(L_G) \neq \text{spec}(L_{G'}), \quad (13)$$

where $L_G = D_G - A_G$ is the combinatorial Laplacian of G , and likewise for G' .

Let $U_G = [u_1, \dots, u_d] \in \mathbb{R}^{|V| \times d}$ be the truncated eigenbasis (top d eigenvectors) of L_G , and similarly $U_{G'}$ for G' . Define spectral embeddings as:

$$X_G^{(0)} = U_G^\top X, \quad X_{G'}^{(0)} = U_{G'}^\top X', \quad (14)$$

where X and X' are initial node features (e.g., degrees or one-hot encodings).

If $\text{spec}(L_G) \neq \text{spec}(L_{G'})$, then:

$$X_G^{(0)} \neq X_{G'}^{(0)}, \quad (15)$$

implying that the node representations in the spectral domain differ.

Further, let the fractal attention mechanism compute attention weights $\alpha_{ij}^{(l)}$ using hierarchical clustering $\mathcal{H}^{(l)}$ over spectral distances:

$$\alpha_{ij}^{(l)} = \text{Softmax} \left(\frac{Q_i^{(l)} K_j^{(l)\top}}{\sqrt{d}} + \mathcal{H}_{ij}^{(l)} \right), \quad (16)$$

which depends on both spectral embeddings and topological structure.

Let the final graph representation be:

$$h_G = \text{MLP} \left(\sum_{l=1}^L \beta_l h_0^{(l)} \right), \quad h_{G'} = \text{MLP} \left(\sum_{l=1}^L \beta_l h_0'^{(l)} \right), \quad (17)$$

where $h_0^{(l)}$ and $h_0'^{(l)}$ are virtual node representations at layer l for G and G' , respectively.

Since:

$$X_G^{(0)} \neq X_{G'}^{(0)} \Rightarrow h_0^{(l)} \neq h_0'^{(l)} \Rightarrow h_G \neq h_{G'}, \quad (18)$$

GraphFractalNet produces different graph embeddings for G and G' , while k -WL produces the same labelings.

Therefore, $\exists G, G'$ s.t. $\text{k-WL}(G) = \text{k-WL}(G')$, but $\text{GraphFractalNet}(G) \neq \text{GraphFractalNet}(G')$, hence proving that GraphFractalNet is strictly more expressive than the k -WL test. $\square$

Theorem 2. Let $\mathcal{H}_F$ be the hypothesis class realized by GraphFractalNet with depth L , spectral embedding dimension d_s , attention/output dimension d_f , and let N denote an upper bound on the number of nodes in graphs of interest. Assume node features satisfy $\|x_i\|_2 \leq B_x$ for all nodes, every linear projection in the network (spectral projection, query/key/value/value-projection, message transforms and MLPs) has operator norm at most B_W , the pointwise nonlinearities are L_σ -Lipschitz, and the loss ℓ is L_ℓ -Lipschitz and bounded in $[0, C_\ell]$. Further, assume the fractal attention mask at each layer is s -sparse per row with $s \leq C \log N$ for some constant $C > 0$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over m i.i.d. training graphs $\{(G_i, y_i)\}_{i=1}^m$, every $h \in \mathcal{H}_F$ satisfies

$$\mathbb{E}[\ell(h(G), y)] \leq \hat{\mathbb{E}}_m[\ell(h(G), y)] + 2L_\ell B_x (B_W L_\sigma)^L \sqrt{\frac{(d_s + d_f)s}{m}} + C_\ell \sqrt{\frac{\log(1/\delta)}{2m}}. \quad (19)$$

Proof. Let $\mathfrak{R}_m(\mathcal{H}_F)$ denote the empirical Rademacher complexity of $\mathcal{H}_F$. From Bartlett–Mendelson Bartlett & Mendelson (2002):

$$\mathbb{E}[\ell(h)] \leq \hat{\mathbb{E}}_m[\ell(h)] + 2\mathfrak{R}_m(\ell \circ \mathcal{H}_F) + C_\ell \sqrt{\frac{\log(1/\delta)}{2m}}. \quad (20)$$

Since ℓ is L_ℓ -Lipschitz,

$$\mathfrak{R}_m(\ell \circ \mathcal{H}_F) \leq L_\ell \mathfrak{R}_m(\mathcal{H}_F). \quad (21)$$

For $x_i \in \mathbb{R}^{d_s + d_f}$, $\|x_i\|_2 \leq B_x$, and any weight matrix W with $\|W\|_2 \leq B_W$,

$$\mathfrak{R}_m(\{x \mapsto Wx\}) \leq B_x B_W \sqrt{\frac{d_s + d_f}{m}}. \quad (22)$$

Fractal attention with at most s nonzeros per row yields $\|A\|_2 \leq \sqrt{s}$, thus

$$\mathfrak{R}_m(\{x \mapsto AWx\}) \leq B_x B_W \sqrt{\frac{(d_s + d_f)s}{m}}. \quad (23)$$

Each nonlinear layer is L_σ -Lipschitz; depth L composition:

$$\mathfrak{R}_m(\mathcal{H}_F) \leq B_x (B_W L_\sigma)^L \sqrt{\frac{(d_s + d_f)s}{m}}. \quad (24)$$

Substitute into the generalization bound

$$\mathbb{E}[\ell(h)] \leq \hat{\mathbb{E}}_m[\ell(h)] + 2L_\ell B_x (B_W L_\sigma)^L \sqrt{\frac{(d_s + d_f)s}{m}} + C_\ell \sqrt{\frac{\log(1/\delta)}{2m}}. \quad (25)$$

□

This theorem quantifies how GraphFractalNet’s architecture and inductive choices control generalization. The bound shows that, under natural norm and Lipschitz constraints, the generalization gap scales inversely with the square root of the number of training graphs m and grows with the model’s effective capacity captured by spectral dimension d_s , attention dimension d_f , depth L , and the per-row sparsity s induced by the fractal mask. Practically, the result explains why spectral truncation (small d_s), sparse fractal attention (small s), and moderate depth help keep the model both expressive and generalizable on large graphs: these design choices reduce the complexity term while preserving multi-scale expressivity, thereby giving a principled trade-off between scalability and learning capacity in GraphFractalNet.

C HOW POWERFUL IS GRAPHFRACTALNET?

GraphFractalNet inherits and extends the expressive capabilities of both spectral graph theory and hierarchical attention mechanisms. By integrating spectral embeddings with fractal attention and dynamic graph rewiring, the model captures multi-scale structural dependencies that cannot be represented by conventional message-passing GNNs limited by the Weisfeiler–Lehman (WL) hierarchy.

From a theoretical perspective, GraphFractalNet can simulate a wide class of existing GNN architectures by appropriately setting its spectral basis functions and attention weights, thereby matching the discriminative power of Graph Isomorphism Networks (GIN) and spectral convolutional methods. Moreover, the spectral encoding inherently encodes global topological information through Laplacian eigenvectors, while fractal attention progressively refines this representation to capture higher-order substructure patterns beyond the reach of k -WL tests for $k \geq 2$.

We formally establish that GraphFractalNet is strictly more expressive than the k -WL test under mild assumptions on the spectral embedding dimensionality and fractal attention depth. Furthermore, leveraging Bartlett–Mendelson Rademacher complexity bounds Bartlett & Mendelson (2002), we derive a generalization guarantee that scales sub-quadratically in the number of nodes N , ensuring that the model remains expressive without overfitting in large-graph regimes. This balance between representational power and theoretical generalization is a key factor in GraphFractalNet’s strong empirical performance across diverse graph learning benchmarks.

D DATASET STATISTICS AND EVALUATION SETTINGS

Table 5 summarizes the datasets used in our experiments along with their statistics and key characteristics. We organize them into three groups based on their source.

- **Graph-level benchmarks:** This includes ZINC, MNIST, CIFAR10, PATTERN, CLUSTER, Peptides-func, and Peptides-struct. For these datasets, we adopt the standard training, validation, and test partitions as well as evaluation measures reported in Rampásek et al. (2022). Further implementation details are also aligned with the protocols from Rampásek et al. (2022).
- **Node-level benchmarks:** This group covers Cora, Citeseer, Pubmed, Actor, Squirrel, Chameleon, ogbn-proteins, ogbn-arxiv, ogbn-products, and ogbn-papers100M. We strictly follow the same data splits and evaluation settings described in Wu et al. (2023) to ensure fair comparison with prior work.
- **Arxiv-year dataset:** This citation network comprises all computer science papers on arXiv, with nodes representing individual papers and edges denoting citation relationships. Each node is described by a 128-dimensional representation derived from averaging Word2Vec embeddings of the paper’s title and abstract. The task is to predict the publication year, discretized into five intervals, using the public 50%/25%/25% train/validation/test split protocol introduced in Lim et al. (2021).

Table 5: Overview of the graph learning dataset.

Dataset	# Graphs	Avg. # nodes	Avg. # edges	# Feats	Prediction level	Prediction task	Metric
CLUSTER	12,000	117.2	2,150.9	7	node	6-class classif.	Accuracy
MNIST	70,000	70.6	564.5	3	graph	10-class classif.	Accuracy
ZINC	12,000	23.2	24.9	28	graph	regression	MAE
PATTERN	14,000	118.9	3,039.3	3	node	binary classif.	Accuracy
CIFAR10	60,000	117.6	941.1	5	graph	10-class classif.	Accuracy
Peptides-struct	15,535	150.9	307.3	9	graph	11-task regression	MAE
Peptides-func	15,535	150.9	307.3	9	graph	10-task classif.	AP
Chameleon	1	2,277	36,101	2,325	node	5-class classif.	Accuracy
Squirrel	1	5,201	216,933	2,089	node	5-class classif.	Accuracy
Actor	1	7,600	26,659	931	node	5-class classif.	Accuracy
Pubmed	1	19,717	44,324	500	node	3-class classif.	Accuracy
Citeseer	1	3,327	4,522	3,703	node	6-class classif.	Accuracy
Cora	1	2,708	5,278	2,708	node	7-class classif.	Accuracy
ogbn-papers100M	1	111,059,956	1,615,685,872	128	node	172-class classif.	Accuracy
ogbn-products	2	2,449,029	61,859,140	100	node	47-class classif.	Accuracy
arxiv-year	1	169,343	1,166,243	128	node	5-class classif.	Accuracy
ogbn-arxiv	1	169,343	1,166,243	128	node	40-class classif.	Accuracy
ogbn-proteins	1	132,534	39,561,252	8	node	112 binary classif.	ROC-AUC

E ADDITIONAL EXPERIMENTAL RESULTS

E.1 RESULTS ON LARGE-SCALE HETEROPHILIC NETWORKS

To further assess the robustness of GraphFractalNet, we extend our evaluation to two challenging large-scale heterophilic benchmarks: Pokec and snap-patents. The Pokec dataset captures a friendship network from a Slovak online social platform, where nodes correspond to users, directed edges indicate friendship relations, and features are derived from profile metadata such as region, registration time, and age. Each node is labeled with the user’s gender. On the other hand, the snap-patents dataset consists of U.S. utility patents, where nodes represent patents and edges encode citation links; node attributes are obtained from patent metadata.

Following the official splits and preprocessing provided in LINKX, we report mean accuracy across five independent runs. As summarized in Table 6, GraphFractalNet consistently surpasses strong baselines such as GOAT, LINKX, CoBFormer, and Exphormer. Notably, it achieves 85.76% accuracy on Pokec and 64.12% on snap-patents, demonstrating its capability to handle large-scale heterophilic graphs more effectively than prior state-of-the-art models.

Table 6: Results on large-scale heterophilic datasets.

	Pokec	snap-patents
	Accuracy $\uparrow$	Accuracy $\uparrow$
GOAT	84.69 ± 0.18	62.43 ± 0.37
LINKX	82.04 ± 0.07	61.95 ± 0.12
CoBFormer	83.42 ± 0.18	61.82 ± 0.69
Exphormer	84.87 ± 0.29	63.56 ± 0.23
GraphFractalNet	85.76 ± 0.45	64.12 ± 0.28

E.2 ARCHITECTURAL SENSITIVITY ANALYSIS

To better understand the impact of architectural choices on GraphFractalNet, we conduct a detailed sensitivity analysis by varying three key parameters: the number of layers (depth), hidden dimension size, and fractal hierarchy depth. These components are central to the expressiveness and efficiency of the model, as they determine the balance between representation power and scalability.

Table 7 summarizes the performance trends on representative datasets (ZINC, CIFAR10, and ogbn-arxiv). We observe the following:

- **Model Depth:** Increasing the number of layers initially improves performance by enhancing feature extraction. However, beyond 8 layers, the gains diminish and in some cases degrade due to over-smoothing, especially in heterophilic datasets.
- **Hidden Dimension:** Larger hidden sizes improve accuracy up to a point (e.g., 256), but excessively wide layers (512) lead to higher memory consumption and slower training without proportional performance improvement.
- **Fractal Hierarchy Depth:** Adding more fractal levels strengthens structural encoding and boosts long-range dependency modeling. However, performance saturates after 3 levels, suggesting diminishing returns at higher depths.

Overall, these results demonstrate that GraphFractalNet achieves the best trade-off with **6–8 layers**, a **hidden size of 256**, and **2–3 fractal levels**, confirming the robustness of our architectural design.

E.3 STRUCTURAL ENCODING SENSITIVITY ANALYSIS

A key component of GraphFractalNet is its ability to capture structural information through fractal-based spectral encodings. To assess the contribution and robustness of this design, we perform a sensitivity analysis by replacing or modifying the structural encoding scheme. Specifically, we compare four variants:

- **No Encoding:** Model trained without explicit structural information (only node attributes).

Table 7: Architectural sensitivity of GraphFractalNet across depth, hidden dimension, and fractal hierarchy levels. Results are reported as mean $\pm$ std over 5 runs. Best results for each setting are highlighted in **bold**.

Setting	Depth (Layers)			Hidden Dim.			Fractal Levels		
	ZINC (MAE $\downarrow$)	CIFAR10 (Acc. $\uparrow$)	ogbn-arxiv (Acc. $\uparrow$)	ZINC	CIFAR10	ogbn-arxiv	ZINC	CIFAR10	ogbn-arxiv
4	0.061 $\pm$ 0.002	74.32 $\pm$ 0.45	71.89 $\pm$ 0.21	0.065 $\pm$ 0.003	73.55 $\pm$ 0.38	71.20 $\pm$ 0.28	0.064 $\pm$ 0.003	74.01 $\pm$ 0.27	71.43 $\pm$ 0.36
6	0.056 $\pm$ 0.002	75.89 $\pm$ 0.51	72.75 $\pm$ 0.24	0.060 $\pm$ 0.002	74.92 $\pm$ 0.40	72.11 $\pm$ 0.23	0.058 $\pm$ 0.002	75.46 $\pm$ 0.39	72.38 $\pm$ 0.31
8	0.052 $\pm$ 0.002	76.81 $\pm$ 0.48	72.21 $\pm$ 0.26	0.056 $\pm$ 0.002	76.15 $\pm$ 0.36	72.02 $\pm$ 0.29	0.053 $\pm$ 0.001	76.60 $\pm$ 0.32	72.10 $\pm$ 0.27
10	0.054 $\pm$ 0.003	76.23 $\pm$ 0.50	72.85 $\pm$ 0.25	0.055 $\pm$ 0.003	75.78 $\pm$ 0.41	72.67 $\pm$ 0.34	0.054 $\pm$ 0.002	76.22 $\pm$ 0.35	72.80 $\pm$ 0.29
12	0.058 $\pm$ 0.004	75.40 $\pm$ 0.53	72.10 $\pm$ 0.28	0.059 $\pm$ 0.003	75.10 $\pm$ 0.42	71.98 $\pm$ 0.30	0.057 $\pm$ 0.003	75.35 $\pm$ 0.41	72.01 $\pm$ 0.32

- **Spectral Laplacian**: Incorporating Laplacian eigenvectors as structural encodings, a widely used baseline in graph transformers.
- **Random Walk PE**: Using random walk-based positional encodings, which capture local diffusion patterns.
- **Fractal Encoding (Proposed)**: Our hierarchical fractal structural encoding, designed to capture both local and global dependencies through self-similar multi-scale patterns.

Table 8 presents the results on three representative datasets: ZINC (graph regression), CIFAR10 (graph classification), and ogbn-arxiv (node classification). The results demonstrate several important findings:

1. Models without explicit structural encodings perform significantly worse, highlighting the necessity of structural priors for effective learning.
2. Laplacian-based and random-walk encodings improve performance, but their gains saturate quickly and are dataset dependent.
3. GraphFractalNet with fractal encoding consistently outperforms the alternatives, indicating its ability to integrate hierarchical structural cues that strengthen generalization across both homophilic and heterophilic settings.

These findings validate the effectiveness of the proposed fractal structural encoding in balancing local neighborhood information with global topological dependencies, contributing significantly to GraphFractalNet’s performance advantage.

Table 8: Sensitivity of GraphFractalNet to different structural encoding strategies. Results are reported as mean $\pm$ std over 5 runs. Best results are in **bold**.

Encoding Strategy	ZINC (MAE $\downarrow$)	CIFAR10 (Acc. $\uparrow$)	ogbn-arxiv (Acc. $\uparrow$)
No Encoding	0.075 $\pm$ 0.004	71.42 $\pm$ 0.48	69.85 $\pm$ 0.34
Spectral Laplacian	0.064 $\pm$ 0.003	74.55 $\pm$ 0.42	72.34 $\pm$ 0.28
Random Walk PE	0.061 $\pm$ 0.002	75.02 $\pm$ 0.40	72.68 $\pm$ 0.29
Fractal Encoding (Proposed)	0.052 $\pm$ 0.002	76.81 $\pm$ 0.48	72.94 $\pm$ 0.42

Table 9: Ablation study of GraphFractalNet components. Results are mean $\pm$ std over 5 runs. Best results are in **bold**.

Model Variant	ZINC (MAE $\downarrow$)	CIFAR10 (Acc. $\uparrow$)	ogbn-arxiv (Acc. $\uparrow$)
w/o Spectral Embeddings	0.070 $\pm$ 0.003	72.11 $\pm$ 0.39	70.14 $\pm$ 0.31
w/o Rewiring	0.066 $\pm$ 0.002	73.45 $\pm$ 0.42	71.02 $\pm$ 0.27
w/o Fractal Attention	0.062 $\pm$ 0.003	74.23 $\pm$ 0.40	71.48 $\pm$ 0.25
Full GraphFractalNet	0.052 $\pm$ 0.002	76.81 $\pm$ 0.48	72.94 $\pm$ 0.42

E.4 ABLATION STUDY

To better understand the contribution of each component in GraphFractalNet, we conduct an extensive ablation study by incrementally removing or modifying its key modules: spectral structural embeddings, dynamic graph rewiring, and fractal attention. Each variant is trained under identical hyperparameter settings, ensuring a fair comparison.

- **w/o Spectral Embeddings:** We remove the Laplacian-based spectral encodings, leaving only raw node features. This tests the importance of structural priors.
- **w/o Rewiring:** We disable dynamic graph rewiring, so message passing is performed only on the original adjacency. This highlights the effect of adaptive topology refinement.
- **w/o Fractal Attention:** We replace the fractal attention with a standard multi-head attention mechanism. This evaluates whether hierarchical multi-scale attention is necessary.
- **Full Model (GraphFractalNet):** Incorporates all three components jointly.

Table 9 reports results on ZINC, CIFAR10, and ogbn-arxiv. Several observations emerge: (1) Removing spectral embeddings leads to a notable drop in performance, especially on node classification, confirming their importance in encoding global structure. (2) Disabling rewiring degrades performance on heterophilic datasets, showing the value of adaptively capturing long-range dependencies. (3) Replacing fractal attention with standard attention reduces accuracy across the board, highlighting the necessity of fractal multi-scale aggregation. (4) The full model achieves the best performance consistently, demonstrating that the components complement each other synergistically.

These ablations confirm that each architectural innovation, spectral embeddings, rewiring, and fractal attention, plays a distinct and complementary role. Their integration leads to the state-of-the-art performance and efficiency of GraphFractalNet.