Transferability of Graph Transformers with Convolutional Positional Encodings

Javier Porras-Valenzuela
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
jporras@seas.upenn.edu

Zhiyang Wang
Halıcıoğlu Data Science Institute
University of California, San Diego
La Jolla, CA 92093
zhw135@ucsd.edu

Xiaotao Shang
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
tshang@seas.upenn.edu

Alejandro Ribeiro
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
aribeiro@seas.upenn.edu

Abstract

Transformers have achieved remarkable success across domains, motivating the rise of Graph Transformers (GTs) as attention-based architectures for graph-structured data. A key design choice in GTs is the use of Graph Neural Network (GNN)-based positional encodings to incorporate structural information. In this work, we establish a theoretical connection between GTs with GNN positional encodings and Manifold Neural Networks (MNNs). Building on transferability results for GNNs, we prove that such GTs inherit the transferability guarantees of GNNs. In particular, GTs trained on small graphs provably generalize to larger graphs under mild assumptions. We complement our theory with extensive experiments on standard graph benchmarks, demonstrating that GTs exhibit scalable generalization behavior on par with GNNs. Our results provide new insights into the understanding of GTs and suggest practical directions for efficient training of GTs in large-scale settings.

1 Introduction

Transformers have recently been adapted to graph-structured data by injecting graph information through positional or structural encodings while retaining global self-attention—yielding Graph Transformers. Graph transformers have delivered state-of-the-art or highly competitive results in several domains, including but not limited to large-scale molecular property prediction [1], biomedical knowledge graphs [2], and long-range data benchmark [3].

Graph transformers extend self-attention to graphs by injecting structural information—via positional or structural encodings—and letting attention aggregate signals beyond local neighbor-

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Graph Machine Learning (NPGML).

hoods. Early formulations introduced absolute encodings from the graph Laplacian and adapted full-attention layers to irregular topologies, while subsequent designs added relative encodings (e.g., shortest-path distance, edge features, centrality) or kernel-based encodings to bias attention toward graph structure. Canonical examples include the Graph Transformer [4], SAN, which learns spectral encodings drawn from the Laplacian spectrum [5], GraphiT (diffusion-kernel/relative encodings) [6], and the Graphormer family (shortest-path and centrality biases in dense self-attention) [1]. Hybrid "local-global" models such as GraphGPS [7], GraphTrans [8] combine a message-passing GNN block with a global attention block, and sparse variants such as Exphormer [9] and , and UnifiedGT [10] replace quadratic attention with structured sparsity for scalability—together yielding a general recipe for high-capacity, size-aware graph transformers.

A central insight across this literature is that structure must be encoded explicitly for attention to be effective on graphs. While full attention is given by transformer architecture, the outstanding performance is always restricted by the huge calculation complexity brought by the full attention architecture. Therefore, it is meaningful to develop a graph transformer that can be transferable across different graph sizes under theoretical guarantees.

Theory from spectral graph signal processing establishes that such graph convolutional filters—under continuity conditions—are stable to perturbations and transferable across graphs sampled from the same limit model [11, 12, 13, 14, 15]. Moreover, by analyzing graphs through limits, one obtains that graph filters and GNNs converge as graph size grows; hence, models trained on small sampled graphs can be deployed on larger graphs from the same limit model without retraining. We show that when these stable and transferable encodings are fed to a transformer whose attention is controlled to be Lipschitz—e.g., by normalization schemes for self-attention or by alternative Lipschitz attention maps—the composed model inherits stability and size-transferability. Practically, this yields an efficient recipe: train on small graphs using graph convolutional positional encodings, then transfer to larger graphs while keeping attention regularized, achieving sub-linear performance difference and substantial computational savings.

The main contributions are as follows:

- We propose the graph convolutional filter as the positional encoding to ensure the stability, equivariance, transferability, and generalization as the transformer inputs.
- With the graph filtering positional encoding, we provide the theoretical guarantee that the graph transformers are transferable across different scales of graphs sampled from an underlying manifold without retraining.
- We carry out experiments to verify on different domains (ArXiv-year, Reddit, snap-patents, MAG). We propose a practical sparse graph transformer whose performance can match or outperform GNNs and other graph transformers especially over heterophilic graphs.

2 Preliminaries

2.1 Graph neural networks

Set up and graph convolutions An undirected graph $\mathbf{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ contains a node set $\mathcal{V}$ with N nodes and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The weight function $\mathcal{W} : \mathcal{E} \to \mathbb{R}$ assigns values to the edges. We define the graph Laplacian $\mathbf{L} = \text{diag}(\mathbf{A1}) - \mathbf{A}$ where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix. Graph signals are functions mapping nodes to a feature value. We write it as a vector $\mathbf{z} \in \mathbb{R}^N$, with each entry $[\mathbf{z}]_i$ representing the function value on node i. A graph convolutional filter $\mathbf{h_G}$ is composed of consecutive graph shifts by graph Laplacian, defined as $\mathbf{h_G}(\mathbf{L})\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{L}^k \mathbf{z}$ with $\{h_k\}_{k=0}^{K-1}$ as filter parameters. We replace $\mathbf{L}$ with eigendecomposition $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$, where $\mathbf{V}$ is the eigenvector matrix and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues $\{\lambda_{i,N}\}_{i=1}^N$ as the entries. The spectral representation of a graph filter is

$$\mathbf{V}^H \mathbf{h_G}(\mathbf{L})\mathbf{z} = \sum_{k=1}^{K-1} h_k \mathbf{\Lambda}^k \mathbf{V}^H \mathbf{z} = \hat{h}(\mathbf{\Lambda})\mathbf{V}^H \mathbf{z}. \tag{1}$$

This leads to a point-wise frequency response of the graph convolution as $\hat{h}(\lambda) = \sum_{k=0}^{K-1} h_k \lambda^k$.

Graph neural networks A graph neural network (GNN) is a layered architecture, where each layer consists of a bank of graph convolutional filters followed by a point-wise nonlinearity $\sigma : \mathbb{R} \to \mathbb{R}$. Specifically, the e-th layer of a GNN that produces F_e output features $\{\mathbf{z}_e^p\}_{p=1}^{F_e}$ with F_{e-1} input features $\{\mathbf{z}_{e-1}^q\}_{q=1}^{F_{e-1}}$ is written as

$$\mathbf{z}_e^p = \sigma\left(\sum_{q=1}^{F_{e-1}} \mathbf{h}_{\mathbf{G}}^{epq}(\mathbf{L})\mathbf{z}_{e-1}^q\right), \tag{2}$$

for each layer $e = 1, 2 \cdots, E$. The graph filter $\mathbf{h}_{\mathbf{G}}^{epq}(\mathbf{L})$ maps the q-th feature of layer $e-1$ to the p-th feature of layer e. We denote the GNN as a mapping $\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}, \mathbf{Z})$, where $\mathcal{H} = \{\mathbf{h}_G^{epq}\}_{e,p,q}$ denotes a set of the graph filter coefficients with a finite dimension at all layers and $\mathbf{Z} \in \mathbb{R}^{N \times F_0}$ as the input feature matrix over all nodes.

2.2 Manifold neural networks

Setup and manifold convolutions. We consider a d-dimensional compact, smooth and differentiable Riemannian submanifold $\mathcal{M}$ embedded in a M-dimensional space $\mathbb{R}^{\mathsf{M}}$ with finite volume. This induces a measure μ which has a non-vanishing Lipschitz continuous density ρ with respect to the Riemannian volume over the manifold with $\rho : \mathcal{M} \to (0, \infty)$, assumed to be bounded as $0 < \rho_{min} \leq \rho(x) \leq \rho_{max} < \infty$ for all $x \in \mathcal{M}$. The manifold data supported on each point $x \in \mathcal{M}$ is defined by scalar functions $f : \mathcal{M} \to \mathbb{R}$ [12]. We use $L^2(\mathcal{M})$ to denote L^2 functions over $\mathcal{M}$ with respect to measure μ. The manifold with probability density function ρ is equipped with a weighted Laplace operator [16], generalizing the Laplace-Beltrami operator as

$$\mathcal{L}f = -\frac{1}{2\rho}\text{div}(\rho^2 \nabla f), \tag{3}$$

with div denoting the divergence operator of $\mathcal{M}$ and ∇ denoting the gradient operator of $\mathcal{M}$ [17, 18]. The manifold convolution operation is defined relying on the Laplace operator $\mathcal{L}$ [12]. For a function $f \in L^2(\mathcal{M})$ as input, a manifold convolutional filter [12] can be defined as

$$g(x) = \mathbf{h}(\mathcal{L})f(x) = \sum_{k=0}^{K-1} h_k e^{-k\mathcal{L}} f(x), \tag{4}$$

with $h_k \in \mathbb{R}$ the filter parameter.

Manifold neural networks. A manifold neural network (MNN) is constructed by cascading L layers, each of which contains a bank of manifold convolutional filters and a pointwise nonlinearity $\sigma : \mathbb{R} \to \mathbb{R}$. The output manifold function of each layer $l = 1, 2 \cdots, L$ can be explicitly denoted as

$$f_l^p(x) = \sigma\left(\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x)\right), \tag{5}$$

where f_{l-1}^q, $1 \leq q \leq F_{l-1}$ is the q-th input feature from layer $l-1$ and f_l^p, $1 \leq p \leq F_l$ is the p-th output feature of layer l. We denote MNN as a mapping $\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, f)$, where $\mathcal{H} = \{\mathbf{h}_l^{pq}\}_{l,p,q}$ is a collective set of filter parameters in all the manifold convolutional filters.

3 Transferable Graph Transformers

We consider signals supported on the manifold defined in Section 2.2, with a weighted Laplace operator as defined in (3). Because functions $f \in L^2(\mathcal{M})$ describe information on $\mathcal{M}$, we focus on a finite-dimensional subspace of $L^2(\mathcal{M})$ determined by an eigenvalue cutoff of $\mathcal{L}$, i.e., a bandlimited signal:

Definition 1. *A manifold signal $f \in L^2(\mathcal{M})$ is bandlimited if there exists some $\lambda > 0$ such that for all eigenpairs $\{\lambda_i, \boldsymbol{\phi}_i\}_{i=1}^{\infty}$ of the weighted Laplacian $\mathcal{L}$ when $\lambda_i > \lambda$, we have $\langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} = 0$.*

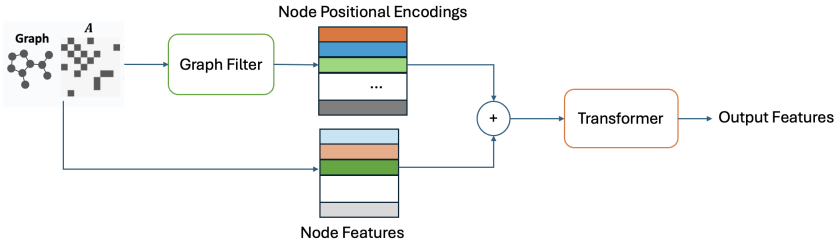


Figure 1: Framework of graph transformer with convolutional filtering positional encodings.

Suppose we are given a set of N i.i.d. randomly sampled points $X_N = \{x_i\}_{i=1}^N$ over $\mathcal{M}$, with $x_i \in \mathcal{M}$ sampled according to measure μ. We construct a graph $\mathbf{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ on these N sampled points X_N, where each point x_i is a vertex of graph $\mathbf{G}$, i.e. $\mathcal{V} = X_N$. Each pair of vertices (x_i, x_j) is connected with an edge while the weight attached to the edge $\mathcal{W}(x_i, x_j)$ is determined by a kernel function K_ϵ. The kernel function is decided by the Euclidean distance $\|x_i - x_j\|$ between these two points. The graph Laplacian denoted as $\mathbf{L}$ can be calculated based on the weight function [19]. The constructed graph Laplacian with an appropriate kernel function has been proved to approximate the Laplace operator $\mathcal{L}$ of $\mathcal{M}$ [20, 21, 22]. In this paper, we implement the normalized Gaussian kernel definition in [22], which is defined as:

$$\mathcal{W}(x_i, x_j) = K_\epsilon(x_i, x_j) = \frac{1}{\epsilon^2} e^{-\frac{\|x_i - x_j\|^2}{4\epsilon^2}}. \tag{6}$$

We consider a graph transformer operating over this constructed graph $\mathbf{G}$ from the underlying manifold $\mathcal{M}$. A graph transformer (GT) is comprised of E GNN layers followed by $L - E$ transformer layers, explicitly denoted as

$$\mathbf{Z}_e = \mathbf{\Psi}_G(\mathcal{H}, \mathbf{L}, \mathbf{Z})_e \qquad e \in [1, E] \tag{7}$$

$$\mathbf{X}_l = \mathbf{\Phi}_G(\mathbf{Z}; \mathbf{T})_l = \mathbf{V}_l \mathbf{X}_{l-1} \text{ softmax}\left[(\mathbf{Q}_l \mathbf{X}_{l-1})^\top (\mathbf{K}_l \mathbf{X}_{l-1})\right] \qquad l \in [E+1, L] \tag{8}$$

with $\mathbf{X}_0 = \mathbf{Z}_E = \mathbf{\Psi}_\mathbf{G}(\mathcal{H}, \mathbf{L}, \mathbf{Z})_E$, and $\mathbf{\Psi}_\mathbf{G}$ a GNN as defined in (2). The outputs of the GNN in Equation (7) are referred to as the positional encodings. The learnable parameters of the transformer are linear maps $\mathbf{Q}_l$,$\mathbf{K}_l$,$\mathbf{V}_l \in \mathbb{R}^{D \times D}$, collected in $\mathbf{T} = \{\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l\}_{l=1}^L$. Observe that in this architecture, the graph structure is only considered in Equation (7). The attention operation (8) computes attention coefficients for every pair of node embeddings $\mathbf{x}_{l,i}, \mathbf{x}_{l,j}$, $i, j \in [1, N]$, regardless of the connectivity in $\mathbf{L}$. The output of the GT is a matrix $\mathbf{X}_L \in \mathbb{R}^{N \times D}$. For the ease of presentation, we consider the case with $E = 1$ and $L = 2$, while the conclusion can be extended to accomodate multiple layers.

Manifold transformer. A manifold transformer layer is defined as

$$f(x) = \mathbf{\Psi}_\mathcal{M}(\mathcal{H}, \mathcal{L}, g)(x) = \sigma\left(\int_\mathcal{M} \tilde{h}(t) e^{-\mathcal{L}t} g(x) d\mu(x)\right) \tag{9}$$

$$\mathbf{\Phi}_\mathcal{M}(\mathbf{T}, f)(x) = \frac{\int_\mathcal{M} e^{\langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle} \mathbf{V} f(y) d\mu(y)}{\int_\mathcal{M} e^{\langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle} d\mu(y)} \tag{10}$$

for manifold signal $g \in L^2(\mathcal{M})$ and $x \in \mathcal{M}$ a point in the manifold. Here, f and g are vector-valued functions over $\mathcal{M}$. Equation (9) corresponds to the MNN described in Section 2.2 over the manifold signal g. Equation (10) describes manifold attention, the continuous analogue of softmax. The manifold transformer is a map $\mathbf{\Phi}_\mathcal{M} : L^2(\mathcal{M}) \to L^2(\mathcal{M})$ resulting of the composition of f (the MNN) with the manifold attention operation. The vector-valued function $\mathbf{\Phi}_\mathcal{M}(\mathbf{T}, f) : \mathcal{M} \to \mathbb{R}^D$ maps a point in the manifold to a D-dimensional signal. We now introduce a set of assumptions required to ensure the convergence from GNNs to MNNs and from GTs to MTs.

Assumption 1 (Normalized Lipschitz signals)**.** *The manifold signals g are normalized Lipschitz for all points $a, b \in \mathcal{M}$, $\|g(b) - g(a)\| \leq \|b - a\|$.*

Assumption 2 (Bounded linear operators)**.** $\mathbf{Q}$*,* $\mathbf{K}$*, and* $\mathbf{V}$ *are bounded linear operators with constants* $C_Q, C_K, C_V > 0$*, i.e.,* $\|\mathbf{Q}\mathbf{x}\| \leq C_Q\|\mathbf{x}\|$*,* $\|\mathbf{K}\mathbf{x}\| \leq C_K\|\mathbf{x}\|$*,* $\|\mathbf{V}\mathbf{x}\| \leq C_V\|\mathbf{x}\|$*, for all* $\mathbf{x} \in \mathbb{R}^D$*.*

Assumption 3 (Spectral continuity of the filter)**.** *The frequency response function of the filter satisfies*

$$|\hat{h}(\lambda)| = \mathcal{O}(\lambda^{-d}), \qquad |\hat{h}'(\lambda)| \leq C_L\lambda^{-d-1}, \quad \lambda \in (0, \infty),$$

with C_L *a spectral continuity constant that regularizes the smoothness of the filter function.*

Assumption 4 (Normalized Lipschitz nonlinearity)**.** *The nonlinearity* σ *is normalized Lipschitz continuous, i.e.,*

$$|\sigma(a) - \sigma(b)| \leq |a - b|, \qquad \sigma(0) = 0.$$

Assumptions 1 — 4 are mild assumptions on the properties of the underlying manifold, manifold signals, and filters, and are common in the analysis of Riemannian manifolds for GNN transferability.

Equation (7) describes the positional encodings of the GT. Leveraging GNNs as positional encodings is a principled choice, supported by the established convergence results of GNNs to MNNs that have led to showing desirable architectural properties. Building on this foundation, we present a convergence theorem from the literature, tailored to our setting.

Theorem 1. *(Point-wise Convergence of GNN to MNNs) For a graph* $\mathbf{G}$ *sampled from a manifold* $\mathcal{M}$ *constructed with Equation 6, for each node* $x_j \in X_N$*, under assumptions 1 — 4, it holds with probability* $1 - \delta$ *that*

$$\Delta_{GNN} = \|[\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}, \mathbf{P}_N f)]_j - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, f)(x_j)\|_2 \leq EF^{E-1}\left(C\epsilon^2 + \frac{\pi^2}{6}\sqrt{\frac{\log 1/\delta}{N}}\right), \tag{11}$$

where C *is a constant that depend on the geometry of the manifold and scales with* C_L*, defined in Appendix 5.2, and* $\epsilon = \epsilon(N) \geq \left(\frac{\log N}{N}\right)^{\frac{1}{2d+12}}$*.*

Having stated the convergence of the GNN positional encodings to the continuous analogue MNN positional encodings, we can use this result to present our main theorem on the convergence of GT with GNN Positional Encodings to MT with MNN positional encodings.

Theorem 2. *(Point-wise Convergence of GT to MT) For any* $x \in \mathcal{M}$*, under assumptions 1 — 2, the pointwise output difference between a graph transformer and manifold transformer, with probability at least* $1 - \delta$*, is bounded by*

$$\begin{aligned}\Delta_{\mathbf{\Phi}(\mathbf{X})} = \|\mathbf{\Phi}_{\mathbf{G}}(\mathbf{T}, \mathbf{X})(x) - \mathbf{\Phi}_{\mathcal{M}}(\mathbf{T}, f)(x)\|_2 \leq (C_V + 2e^M C_{QK})\Delta_{GNN} \\ + \left[(C_V + e^M C_{QK})\right] A(\frac{\log N}{N})^{1/d}\end{aligned} \tag{12}$$

where A *is a constant related to the geometry of* $\mathcal{M}$*,* $d \geq 3$ *is the intrinsic dimension of the manifold,* $C_{QK} = C_Q C_K$*,* C_V *are the linear operator bound constants of* $\mathbf{Q}$*,* $\mathbf{K}$*, and* $\mathbf{V}$*, and* $M = C_{QK}$*.*

The proof of Theorem 2 is available in Appendix 7. This result proves that as the number of nodes sampled in graph $\mathbf{G}$ increases, the output of GT tends to converge to the underlying MT with a rate $\mathcal{O}\left((\log N/N)^{(1/d)}\right)$. The Lipschitz constant of the value operator appears in the term $C_V\Delta_{GNN}$, which indicates smoother graph filters in GNNs lead to a smaller convergence bound. Furthermore, the result suggests smoother linear operators $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{H}$ can also improve the convergence rate, which provides the insight that by adding regularization to the operators in transformer helps to achieve a better convergence result, hence a better transferability performance.

Theorem 2 indicates that the pointwise output difference of GT and MT decays as the size of the sampled graph N grows. This implies that we can train a GT on a small graph $\mathbf{G}_1$ with N_1 nodes, and *transfer* it to a larger graph $\mathbf{G}_2$, with $N_2 < N_1$, with guarantees that the approximation gap to the manifold transformer's output is bounded. This implication is paramount given the $\mathcal{O}(N^2)$ computational cost of GT – we can train on a relatively smaller graph and ensure good performance on larger graphs. We state this below in the following corollary:

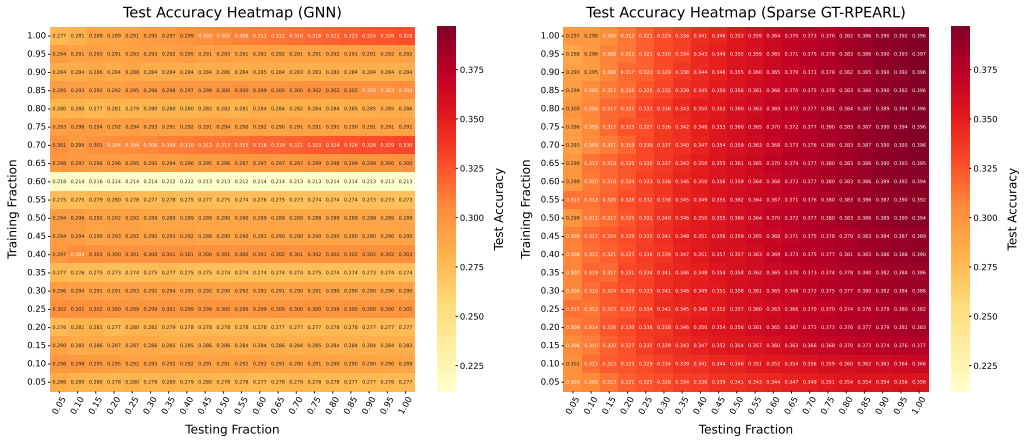


Figure 2: Test Accuracy Heatmaps on snap-patents across for GCN and SGT-RPEARL.

Corollary 3. *(Transferability of Graph Transformers) Let* $\mathbf{G}_1$ *and* $\mathbf{G}_2$ *graphs constructed by points sampled from* $\mathcal{M}$*, with* N_1*,* N_2 *nodes respectively, and graph signals* $\mathbf{X}_1$ *and* $\mathbf{X}_2$*. Further, let* $\mathbf{I}_N : L_2(\mathbf{G}_N) \to L_2(\mathcal{M})$ *denote the interpolation operator on* N *nodes (detailed definition in Appendix 5.1). Define the* $1,2$ *norm of a manifold signal* f *supported on* $\mathcal{M}$ *as* $||f||_{L^{1,2}(\mathcal{M})} = \int_{\mathcal{M}} \|f(x)\|_2 \, d\mu(x)$ *Then, it holds, with probability* $1-\delta$*,*

$$\frac{1}{\mu(\mathcal{M})}||\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{T},\mathbf{X}_1) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathbf{T},\mathbf{X}_2)||_{L^{1,2}(\mathcal{M})} \leq \Delta_{\mathbf{\Phi}(\mathbf{X}_1)} + \Delta_{\mathbf{\Phi}(\mathbf{X}_2)} + 2C_V C_{QK} r \quad (13)$$

4 Experiments

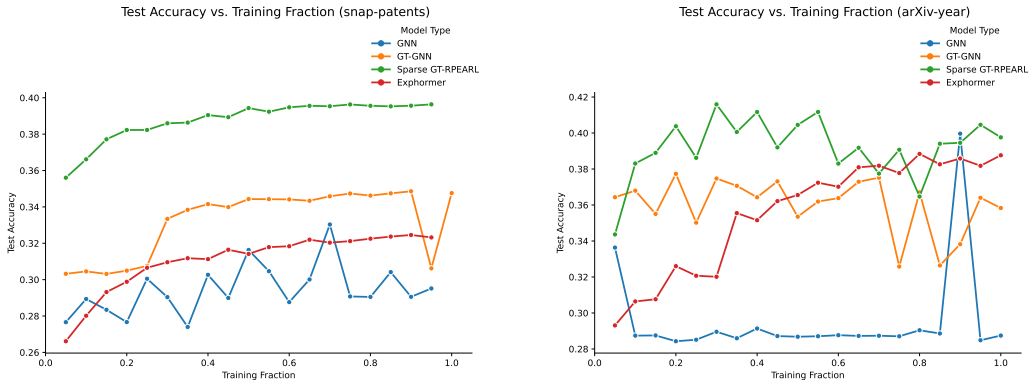
Corollary 3 implies that the performance gap of GTs versus the ideal manifold transformer should decay as graph sizes increase, a consequence that we now turn to validating empirically. From the dataset, we subsample training graphs G_{TR} with sizes N_{TR} taken over fractions $0.05, 0.1, \ldots, 1.0$, and evaluate on a large test graph G_{TST} with size $N_{\text{TST}} \gg N_{\text{TR}}$. Our theory predicts that as N_{TR} increases, the performance of GTs on G_{TST} approximates that of a GT trained on the full graph.

We evaluate on four node classification datasets. Here we present SNAP-Patents [23] and ArXiv-year [23]. Results for ogbn-mag [24] and REDDIT-BINARY [25] are available in Appendix 10, which show similar transferability patterns in cases where GCN and GT's accuracy is comparable.

Models. We consider a GCN [26] baseline, and three different transformers. A conventional transformer [27] with attention coefficients for each pair of nodes in the graph and RPEARL [28] positional encodings (GT-GNN). A sparse transformer with attention restricted to the k-hop neighborhoods, with RPEARL-based positional encodings (Sparse GT-RPEARL). Finally, Exphormer [9], another sparse variant that computes attention coefficients for (i) one-hop neighbors, (ii) random edges from an expander graph, (iii) N additional attention coefficients between each node and a virtual global node. Sparse GT and Exphormer are transformer variants that endow the architecture with additional locality and sparsity priors that go beyond our theory to aid with performance and computational tractability on large graphs. Crucially, Exphormer has no positional encodings, and thus is not covered by our theoretical results.

GTs with GCN encodings exhibit transferability properties. Figure 3 shows the test performance of the four models with increasing training fractions on full testing datasets, for SNAP-Patents and ArXiv-year. GT-GNN and SGT-GNN's accuracy with only a fraction of the training nodes is comparable to their accuracy with the largest training fraction, indicating successful transferability. This is also true for GCN, albeit with a lower peak accuracy. This gap between GCN and GT is consistent with previous work's observations of the success of global attention in heterophilic datasets ([4]). The trend of Exphormer in ArXiV-year also shows a more pronounced monotonic increase, possibly due to the random sampling procedure being more beneficial on larger graphs.

GTs attain high accuracy on small test graphs. The heatmaps of Figure 2 show the performance of each model with both increasing sizes of train and test graphs on SNAP-Patents. Here, Sparse GT, maintains strong accuracy even with the smallest possible test graph fractions. In contrast, GCN's accuracy is low with small test graphs, suggesting that it requires a minimum amount of graph structure to make reliable predictions.



(a) snap-patents (b) arXiv-year

Figure 3: Transferability results for arXiv-year and snap-patents.

References

[1] 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.

[2] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

[3] Vijay Prakash Dwivedi, Ladislav Rampášek, 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, 2022.

[4] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[5] 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.

[6] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

[7] 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.

[8] Zhanghao Wu, Paras Jain, Matthew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing Long-Range Context for Graph Neural Networks with Global Attention.

[9] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pages 31613–31632. PMLR, 2023.

[10] Junhong Lin, Xiaojie Guo, Shuaicheng Zhang, Dawei Zhou, Yada Zhu, and Julian Shun. UnifiedGT: Towards a Universal Framework of Transformers in Large-Scale Graph Learning. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1057–1066. IEEE.

[11] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Geometric graph filters and neural networks: Limit properties and discriminability trade-offs. *IEEE Transactions on Signal Processing*, 2024.

[12] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Stability to deformations of manifold filters and manifold neural networks. *IEEE Transactions on Signal Processing*, pages 1–15, 2024.

[13] Luana Ruiz, Luiz FO Chamon, and Alejandro Ribeiro. Transferability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 71:3474–3489, 2023.

[14] Luana Ruiz, Luiz Chamon, and Alejandro Ribeiro. Graphon neural networks and the transferability of graph neural networks. *Advances in Neural Information Processing Systems*, 33:1702–1712, 2020.

[15] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. Convergence and stability of graph convolutional networks on large random graphs. *Advances in Neural Information Processing Systems*, 33:21512–21523, 2020.

[16] Alexander Grigor'yan. Heat kernels on weighted manifolds and applications. *Cont. Math*, 398(2006):93–191, 2006.

[17] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[18] Gal Gross and Eckhard Meinrenken. *Manifolds, vector fields, and differential forms: an introduction to differential geometry*. Springer Nature, 2023.

[19] Russell Merris. A survey of graph Laplacians. *Linear and Multilinear Algebra*, 39(1-2):19–31, 1995.

[20] Jeff Calder and Nicolas Garcia Trillos. Improved spectral convergence rates for graph Laplacians on ε-graphs and k-NN graphs. *Applied and Computational Harmonic Analysis*, 60:123–175, 2022.

[21] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

[22] David B Dunson, Hau-Tieng Wu, and Nan Wu. Spectral convergence of graph Laplacian and heat kernel reconstruction in L^∞ from random samples. *Applied and Computational Harmonic Analysis*, 55:282–336, 2021.

[23] 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. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, pages 20887–20902. arXiv.

[24] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft Academic Graph: When experts are not enough. 1(1):396–413.

[25] Pinar Yanardag and S.V.N. Vishwanathan. Deep Graph Kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1365–1374. Association for Computing Machinery.

[26] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010. Curran Associates Inc.

[28] Charilaos Kanatsoulis, Evelyn Choi, Stefanie Jegelka, Jure Leskovec, and Alejandro Ribeiro. Learning Efficient Positional Encodings with Graph Neural Networks.

[29] Nicolas Garcia Trillos, Moritz Gerlach, Matthias Hein, and Dejan Slepcev. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs towards the Laplace–Beltrami operator.

[30] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[31] Wolfgang Arendt, Robin Nittka, Wolfgang Peter, and Frank Steiner. Weyl's law: Spectral properties of the Laplacian in mathematics and physics. *Mathematical analysis of evolution, information, and complexity*, pages 1–71, 2009.

[32] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro. A Manifold Perspective on the Statistical Generalization of Graph Neural Networks.

5 Appendix

5.1 Manifold decomposition and induced manifold signal

The graph $\mathbf{G}$ contains points $X_N = \{x_i\}_{i=1}^N$ sampled from the manifold $\mathcal{M}$. The graph signal $\mathbf{X} \in \mathbb{R}^{N\times D}$ can be viewed as a discretization of the continuous manifold signal f evaluated at points X_N, that is

$$\mathbf{P}_N f = \mathbf{X}, \tag{14}$$

where $\mathbf{P}_N : L_2(\mathcal{M}) \to L_2(X_N)$ is called the sampling operator. The sample X_N induces a decomposition of the manifold [29], $\{V_i\}_{i=1}^N$ with respect to X_N, with $V_i \subset B_r(x_i)$ a ball of radius r centered at x_i, with respect to the Euclidean distance in Euclidean ambient space.

Let $\mu_N = \frac{1}{N}\sum_{i=1}^N \delta x_i$ the empirical measure of the random sample. The decomposition is defined by the ∞-optimal transport map $T : \mathcal{M} \to X_N$, defined by the ∞-optimal Transport Distance between μ and μ_N,

$$d_\infty(\mu, \mu_N)L = \min_{T:T\#\mu=\mu_N} \text{ess sup}_{x\in\mathcal{M}}\delta(x, t(x)). \tag{15}$$

Here, $T\#\mu$ denotes that $\mu(T^{-1}(V)) = \mu_N(V)$ holds for every V_i of the decomposition of $\mathcal{M}$.

The radius of the balls where the partitions are contained can be bounded as $r \leq A(\frac{\log N}{N})^{1/d}$ when $d \geq 3$ and as $r \leq A(\log N)^{3/4}/N^{1/2}$ when $d = 2$, with A being a constant related to the geometry of the manifold.

The manifold function induced by the signals of the sampled graph is a piecewise constant function defined by

$$(\mathbf{I}_N\mathbf{X})(x) = \sum_{i=1}^N [\mathbf{X}]_i \mathbb{1}_{x\in V_i} \tag{16}$$

where $\mathbf{I}_N : L_2(X_N) \to L_2(\mathcal{M})$ denotes the interpolation operator.

5.2 Proof of Theorem 1

We first import the spectral point-wise convergence of graph Laplacian to Laplace-Beltrami operator from [22]. The spectral representation of manifold filters are similar to graph convolutional filter, while we consider the case in which the Laplace operator is self-adjoint, positive-semidefinite and the manifold $\mathcal{M}$ is compact. In this case, $\mathcal{L}_\rho$ has real, positive and discrete eigenvalues $\{\lambda_i\}_{i=1}^\infty$, written as $\mathcal{L}_\rho\boldsymbol{\phi}_i = \lambda_i\boldsymbol{\phi}_i$ where $\boldsymbol{\phi}_i$ is the eigenfunction associated with eigenvalue λ_i. The eigenvalues are ordered in increasing order as $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots$, and the eigenfunctions are orthonormal and form an eigenbasis of $L^2(\mathcal{M})$. When mapping a manifold signal onto the eigenbasis $[\hat{f}]_i = \langle f, \boldsymbol{\phi}_i\rangle_\mathcal{M} = \int_\mathcal{M} f(x)\boldsymbol{\phi}_i(x)\mathrm{d}\mu(x)$, the manifold convolution can be seen in the spectral domain as

$$[\hat{g}]_i = \sum_{k=0}^{K-1} h_k e^{-k\lambda_i}[\hat{f}]_i. \tag{17}$$

Hence, the frequency response of manifold filter is given by $\hat{h}(\lambda) = \sum_{k=0}^{K-1} h_k e^{-k\lambda}$.

Proposition 1. *[22][Theorem 4] For a sufficiently small $\epsilon > 0$, if n is sufficiently large so that $\epsilon = \epsilon(n) \geq \left(\frac{\log n}{n}\right)^{\frac{1}{2d+12}}$, then with probability greater than $1 - n^{-2}$, for all $0 \leq i < M$,*

$$|\lambda_{i,N} - \lambda_i| \leq \Omega_1\epsilon^2, \max_{x_j\in X_N} |a_i[\boldsymbol{\phi}_{i,n}]_j - \boldsymbol{\phi}_i(x_j)| \leq \Omega_2\epsilon^2, \tag{18}$$

with Ω_1 and Ω_2 related to the eigengap of $\mathcal{L}$, d, and the diameter, the volume, the injectivity radius, the curvature and the second fundamental form of the manifold.

Because $\{x_1, x_2, \cdots, x_N\}$ is a set of randomly sampled points from $\mathcal{M}$, based on Theorem 19 in [30] we can claim that

$$|\langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}}| = O\left(\sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{19}$$

where $\langle f, \boldsymbol{\phi}_i \rangle = \int_{\mathcal{M}} f(x)\boldsymbol{\phi}_i(x)\mathrm{d}\mu(x)$ is defined as the inner product over manifold $\mathcal{M}$. This also indicates that

$$\left| \|\mathbf{P}_N f\|^2 - \|f\|^2_{\mathcal{M}} \right| = O\left(\sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{20}$$

which indicates $\|\mathbf{P}_N f\| = \|f\|_{\mathcal{M}} + O((\log(1/\delta)/N)^{1/4})$, where $\|f\|^2_{\mathcal{M}} = \langle f, f \rangle_{\mathcal{M}}$. We suppose that the input manifold signal is λ_M-bandlimited with M spectral components. We first write out the difference on each node $x_j \in X_N$ as

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{P}_N f]_j - (\mathbf{h}(\mathcal{L}_\rho)f)(x_j)\| = \left\| \sum_{i=1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{M} \hat{h}(\lambda_i) \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| \tag{21}$$

$$\leq \left\| \sum_{i=1}^{M} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{M} \hat{h}(\lambda_i) \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) + \sum_{i=M+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \tag{22}$$

$$\leq \left\| \sum_{i=1}^{M} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{M} \hat{h}(\lambda_i) \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| + \left\| \sum_{i=M+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\|. \tag{23}$$

The first part of (23) can be decomposed with the triangle inequality as

$$\begin{aligned} &\left\| \sum_{i=1}^{M} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{M} \hat{h}(\lambda_i) \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| \\ &\leq \left\| \sum_{i=1}^{M} \left(\hat{h}(\lambda_{i,N}) - \hat{h}(\lambda_i) \right) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| + \left\| \sum_{i=1}^{M} \hat{h}(\lambda_i) \left(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\|. \end{aligned} \tag{24}$$

In (24), the first part relies on the difference of eigenvalues and the second part depends on the eigenvector difference. The first term in (24) is bounded with Cauchy-Schwartz inequality as

$$\left\| \sum_{i=1}^{M} (\hat{h}(\lambda_{i,n}) - \hat{h}(\lambda_i)) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \leq \sum_{i=1}^{M} \left| \hat{h}(\lambda_{i,N}) - \hat{h}(\lambda_i) \right| |\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle| \tag{25}$$

$$\leq \|\mathbf{P}_N f\| \sum_{i=1}^{M} |\hat{h}'(\lambda_i)| |\lambda_{i,N} - \lambda_i| \tag{26}$$

$$\leq \|\mathbf{P}_N f\| \sum_{i=1}^{M} C_L \Omega_1 \epsilon^2 \lambda_i^{-d} \tag{27}$$

$$\leq \|\mathbf{P}_N f\| C_L \Omega_1 \epsilon^2 \sum_{i=1}^{M} i^{-2} \tag{28}$$

$$\leq \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \Omega_1 \epsilon^2 \frac{\pi^2}{6} := A_1(N) \tag{29}$$

In (25), it depends on the inequality that $|[\boldsymbol{\phi}_{i,N}]_j| \leq \|\boldsymbol{\phi}_{i,N}\|_\infty \leq \|\boldsymbol{\phi}_{i,N}\|_2 = 1$. In (27), it depends on the filter assumption in Assumption 3. In (28), we implement Weyl's law [31] which indicates that eigenvalues of Laplace operator scales with the order $\lambda_i \sim i^{2/d}$. The last inequality comes from

the fact that $\sum_{i=1}^{\infty} i^{-2} = \frac{\pi^2}{6}$. The second term in (24) can be bounded with the triangle inequality as

$$\begin{aligned}
&\left\| \sum_{i=1}^{M} \hat{h}(\lambda_i) \left(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\leq \left\| \sum_{i=1}^{M} \hat{h}(\lambda_i) \left(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\qquad + \left\| \sum_{i=1}^{M} \hat{h}(\lambda_i) \left(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\| \qquad (30)
\end{aligned}$$

The first term in (30) can be bounded with inserting the eigenfunction convergence result in Proposition 1 as

$$\begin{aligned}
&\left\| \sum_{i=1}^{M} \hat{h}(\lambda_i) \left(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\leq \sum_{i=1}^{M} \left| \hat{h}(\lambda_i) \right| \left\| \mathbf{P}_N f \right\| \left| [\boldsymbol{\phi}_{i,N}]_j - \boldsymbol{\phi}_i(x_j) \right| \qquad (31) \\
&\leq \sum_{i=1}^{M} (\lambda_i^{-d+1}) \Omega_2 \epsilon^2 \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \qquad (32) \\
&\leq \Omega_2 \epsilon^2 \sum_{i=1}^{M} (\lambda_i^{-d+1}) \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \qquad (33) \\
&:= A_2(M, N). \qquad (34)
\end{aligned}$$

Considering the filter assumption in Assumption 3, the second term in (30) can be written as

$$\begin{aligned}
&\left\| \sum_{i=1}^{M} \hat{h}(\lambda_{i,N}) (\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j)) \right\| \\
&\leq \sum_{i=1}^{M} \left| \hat{h}(\lambda_{i,N}) \right| \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \left| \boldsymbol{\phi}_i(x_j) \right| \qquad (35) \\
&\leq \sum_{i=1}^{M} (\lambda_{i,N}^{-d}) \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \left\| \boldsymbol{\phi}_i \right\| \qquad (36) \\
&\leq \sum_{i=1}^{M} \left(1 + \Omega_1 \epsilon^2 \right)^{-d} \lambda_i^{-d} \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (37) \\
&\leq \frac{\pi^2}{6} \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| := A_3(N) \qquad (38)
\end{aligned}$$

The term $|\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}}|$ can be decomposed by inserting a term $\langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle$ as

$$\begin{aligned}
&\left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \leq \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle + \langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (39) \\
&\leq \left| \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle - \langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle \right| + \left| \langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (40) \\
&\leq \left\| \mathbf{P}_N f \right\| \left\| \boldsymbol{\phi}_{i,N} - \mathbf{P}_N \boldsymbol{\phi}_i \right\| + \left| \langle \mathbf{P}_N f, \mathbf{P}_N \boldsymbol{\phi}_i \rangle - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (41) \\
&\leq \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \frac{C_{\mathcal{M},2} \lambda_i \sqrt{\epsilon}}{\theta_i} + \sqrt{\frac{\log(1/\delta)}{N}} \qquad (42)
\end{aligned}$$

Then equation (37) can be bounded as

$$\left\| \sum_{i=1}^{M} \hat{h}(\lambda_{i,N})(\langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) - \langle f, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j)) \right\|$$

$$\leq \sum_{i=1}^{M} (1+\Omega_1 \epsilon^2)^{-d} (\lambda_i^{-d}) \left(\left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \frac{C_{\mathcal{M},2} \lambda_i \epsilon}{\theta_i} + \sqrt{\frac{\log(1/\delta)}{N}} \right) \tag{43}$$

$$\leq \frac{\pi^2}{6} \max_{i=1,\cdots,M} \frac{C_{\mathcal{M},2}\epsilon}{\theta_i} \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) + \frac{\pi^2}{6} \sqrt{\frac{\log(1/\delta)}{N}} \tag{44}$$

The second term in (23) can be bounded with the eigenvalue difference bound in Proposition 1 as

$$\left\| \sum_{i=M+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{P}_N f, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \leq \sum_{i=M+1}^{N} (\lambda_{i,N}^{-d}) \left(\|f\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \tag{45}$$

$$\leq \sum_{i=M+1}^{\infty} (\lambda_{i,N}^{-d}) \|f\|_{\mathcal{M}} \tag{46}$$

$$\leq \left(1+\Omega_1 \epsilon^2\right)^{-d} \sum_{i=M+1}^{\infty} (\lambda_i^{-d}) \|f\|_{\mathcal{M}} \tag{47}$$

$$\leq M^{-1} \|f\|_{\mathcal{M}} := A_4(M). \tag{48}$$

We note that the bound is made up by terms $A_1(N) + A_2(M,N) + A_3(N) + A_4(M)$, related to the bandwidth of manifold signal M and the number of sampled points N. This makes the bound scale with the order

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{P}_N f]_j - \mathbf{h}(\mathcal{L}_\rho) f(x_j)\| \leq C_1' \epsilon^2 + C_2' \epsilon \theta_M^{-1} + C_3' \sqrt{\frac{\log(1/\delta)}{N}} + C_4' M^{-1}, \tag{49}$$

with $C_1' = C_L \Omega_1 \frac{\pi^2}{6} \|f\|_{\mathcal{M}}$, $C_2' = \Omega_2 \frac{\pi^2}{6}$, $C_3' = \frac{\pi^2}{6}$ and $C_4' = \|f\|_{\mathcal{M}}$. As N goes to infinity, for every $\delta > 0$, there exists some M_0, such that for all $M > M_0$ it holds that $A_4(M) \leq \delta/2$. There also exists n_0, such that for all $N > n_0$, it holds that $A_1(N) + A_2(M_0, N) + A_3(N) \leq \delta/2$. We can conclude that the summations converge as N goes to infinity. We see M large enough to have $M^{-1} \leq \delta'$, which makes the eigengap θ_M also bounded by ϵ. We combine the first two terms as

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{P}_N f]_j - \mathbf{h}(\mathcal{L}_\rho) f(x_j)\| \leq (C_1 C_L + C_2)\epsilon^2 + \frac{\pi^2}{6} \sqrt{\frac{\log(1/\delta)}{N}}, \tag{50}$$

with $C_1 = \Omega_1 \frac{\pi^2}{6} \|f\|_{\mathcal{M}}$ and $C_2 = \Omega_2 \frac{\pi^2}{6} \theta_{\delta'^{-1}}^{-1}$. To bound the output difference of MNNs, we need to write in the form of features of the final layer

$$\|[\boldsymbol{\Psi}_{\mathbf{G}}(\mathbf{H}, \mathbf{L}_N, \mathbf{P}_N f)]_j - \boldsymbol{\Psi}(\mathbf{H}, \mathcal{L}_\rho, f)(x_j)\| = \left\| \sum_{q=1}^{F} [\mathbf{x}_{n,L}^q]_j - \sum_{q=1}^{F} f_L^q(x_j) \right\| \tag{51}$$

$$\leq \sum_{q=1}^{F} \left\| [\mathbf{x}_{n,L}^q]_j - f_L^q(x_j) \right\|. \tag{52}$$

By inserting the definitions, we have

$$\left\| [\mathbf{x}_{n,l}^p]_j - f_l^p(x_j) \right\| = \left\| \sigma \left(\left[\sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{n,l-1}^q \right]_j \right) - \sigma \left(\sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathcal{L}_\rho) f_{l-1}^q(x_j) \right) \right\| \tag{53}$$

with $\mathbf{x}_{n,0} = \mathbf{P}_N f$ as the input of the first layer. With a normalized point-wise Lipschitz nonlinearity, we have

$$\|[\mathbf{x}_{n,l}^p]_j - f_l^p(x_j)\| \leq \left\| \sum_{q=1}^{F} \left[\mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{n,l-1}^q \right]_j - \sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathcal{L}_\rho) f_{l-1}^q(x_j) \right\| \tag{54}$$

$$\leq \sum_{q=1}^{F} \left\| [\mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{n,l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L}_\rho) f_{l-1}^q(x_j) \right\| \tag{55}$$

The difference can be further decomposed as

$$\begin{aligned}
&\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{n,l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L}_\rho)f_{l-1}^q(x_j)\| \\
&\leq \|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{n,l-1}^q]_j - [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{P}_N f_{l-1}^q]_j + [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{P}_N f_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L}_\rho)f_{l-1}^q(x_j)\| \quad (56)\\
&\leq \left\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{n,l-1}^q]_j - [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{P}_N f_{l-1}^q]_j\right\| + \left\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{P}_N f_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L}_\rho)f_{l-1}^q(x_j)\right\| \quad (57)
\end{aligned}$$

The second term can be bounded with (49) and we denote the bound as Δ_N for simplicity. The first term can be decomposed by Cauchy-Schwartz inequality and non-amplifying of the filter functions as

$$\left\|[\mathbf{x}_{n,l}^p]_j - f_l^p(x_j)\right\| \leq \sum_{q=1}^F \Delta_N \|\mathbf{x}_{n,l-1}^q\| + \sum_{q=1}^F \|[\mathbf{x}_{l-1}^q]_j - f_{l-1}^q(x_j)\|. \quad (58)$$

To solve this recursion, we need to compute the bound for $\|\mathbf{x}_l^p\|$. By normalized Lipschitz continuity of σ and the fact that $\sigma(0) = 0$, we can get

$$\|\mathbf{x}_l^p\| \leq \left\|\sum_{q=1}^F \mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{l-1}^q\right\| \leq \sum_{q=1}^F \|\mathbf{h}_l^{pq}(\mathbf{L}_N)\|\,\|\mathbf{x}_{l-1}^q\| \leq \sum_{q=1}^F \|\mathbf{x}_{l-1}^q\| \leq F^{l-1}\|\mathbf{x}\|. \quad (59)$$

Insert this conclusion back to solve the recursion, we can get

$$\left\|[\mathbf{x}_{n,l}^p]_j - f_l^p(x_j)\right\| \leq lF^{l-1}\Delta_N\|\mathbf{x}\|. \quad (60)$$

Replace l with L we can obtain

$$\|[\mathbf{\Psi}_\mathbf{G}(\mathbf{H}, \mathbf{L}_N, \mathbf{P}_N f)]_j - \mathbf{\Psi}(\mathbf{H}, \mathcal{L}_\rho, f)(x_j)\| \leq LF^{L-1}\Delta_N, \quad (61)$$

when the input graph signal is normalized. By replacing $f = \mathbf{I}_N\mathbf{x}$, we can conclude the proof.

5.3 Local Lipschitz continuity of MNNs

We utilize Proposition 3 in [32], which shows that the outputs of MNN defined in (5) are locally Lipschitz continuous within a certain area, which is stated explicitly as follows.

Proposition 2. *(Local Lipschitz continuity of MNNs [32][Proposition 3]) Assume that the assumptions in Theorem 1 hold. Let MNN be L layers with F features in each layer, suppose the manifold filters are nonamplifying with $|\hat{h}(\lambda)| \leq 1$ and the nonlinearities normalized Lipschitz continuous, then there exists a constant C' such that*

$$|\mathbf{\Phi}(\mathbf{H}, \mathcal{L}_\rho, f)(x) - \mathbf{\Phi}(\mathbf{H}, \mathcal{L}_\rho, f)(y)| \leq F^L C' dist(x - y), \quad \textit{for all } x, y \in B_r(\mathcal{M}), \quad (62)$$

where $B_r(\mathcal{M})$ is a ball with radius r over $\mathcal{M}$ with respect to the geodesic distance.

6 Lemmas and Propositions

Lemma 4. *Let $x \in \mathcal{M}$ a point in the manifold, and $y \in V_j$, a point in partition $V_j \subset B_r(x_j)$. Then it holds that*

$$|\langle \mathbf{Q}f(x), \mathbf{K}f(x_j)\rangle - \langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle| \leq B_f C_Q C_K\, r \quad (63)$$

Proof.

$$\begin{aligned}
|\langle \mathbf{Q}f(x), \mathbf{K}f(x_j)\rangle - \langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle| &= |\langle \mathbf{Q}f(x), \mathbf{K}\left(f(x_j) - f(y)\right)\rangle| && (64)\\
&\leq C_Q\|f(x)\| - C_K\|f(x_j) - f(y)\| && (65)\\
&\leq B_f C_Q C_K \|f(x_j) - f(y)\| && (66)\\
&\leq B_f C_Q C_K |x_j - y| && (67)\\
&\leq B_f C_Q C_K\, r && (68)\\
& && (69)
\end{aligned}$$

Where in (65) we apply the bound on the linear operators $\mathbf{Q}$ and $\mathbf{K}$, in (66) we apply the bound on the manifold signal $\|f(x)\| \leq B$, in (67) we apply the assumption on normalized Lipschitz MNN, $\|f(x) - f(y)\| \leq |x - y|$, and in (68) we use the fact that $y \in V_j$, therefore $|y - x_j| \leq r$. □

Lemma 5. *Let $X_N = \{x_i\}_{i=1}^N$ be a set of points sampled from the manifold $\mathcal{M}$, with its corresponding induced partitioning $\{V_i\}_{i=1}^N$. For each $x_j \in X_N$, and for any $y \in V_j$, it holds that*

$$|\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| \leq e^M B_f C_Q C_K \, r \tag{70}$$

Proof.

$$|\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| = |\exp\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle - \exp\langle \mathbf{Q}f(x_j), \mathbf{K}f(y)\rangle| \tag{71}$$

$$\leq e^M \left[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle - \langle \mathbf{Q}f(x_j), \mathbf{K}f(y)\rangle\right] \tag{72}$$

$$\leq e^M B_f C_Q C_K \, r \tag{73}$$

where $M := \sup_{u,v\in\mathcal{M}} \langle \mathbf{Q}f(u), \mathbf{K}f(v)\rangle$. Note that $M \leq C_{QK} B_f^2$ using the bounds on the MNN signal and linear operators. In the first inequality we use the mean value theorem, $|e^a - e^b| \leq e^{\max\{a,b\}}|a-b|$. In the second we apply the bound from Lemma 4. □

Lemma 6. *Let $X_N = \{x_i\}_{i=1}^N$ be a set of points sampled from the manifold $\mathcal{M}$, with its corresponding induced partitioning $\{V_i\}_{i=1}^N$. For each $x_j \in X_N$ it holds that*

$$|\gamma_{ij} - \tilde{\gamma}_{ij}| = \left|\exp\left[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle\right] - \exp\left[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle\right]\right| \tag{74}$$

$$\leq e^M \left|\left[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle\right] - \left[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle\right]\right| \tag{75}$$

$$\leq e^M C_{QK} \left|\langle \mathbf{x}_i, \mathbf{x}_j\rangle - \langle f(x_i), f(x_j)\rangle\right| \tag{76}$$

$$\leq e^M C_{QK} \left|\langle \mathbf{x}_i, (\mathbf{x}_j - f(x_j)\rangle + \langle (\mathbf{x}_i - f(x_i)), f(x_j)\rangle\right| \tag{77}$$

$$\leq e^M C_{QK} \left[\|\mathbf{x}_i - f(x_i)\| \cdot \|\mathbf{x}_j\| + \|f(x_i)\| \|\mathbf{x}_j - f(x_j)\|\right] \tag{78}$$

$$\leq 2e^M C_{QK} B_f \Delta_{GNN} \tag{79}$$

$$\leq 2e^M C_{QK} \Delta_{GNN}. \tag{80}$$

In equation (75) we apply the mean value theorem, in (76) we apply the bound on the linear operators, in (77) add and subtract an intermediate term and apply bilinearity of inner products, in (78) we apply Cauchy-Schwartz, in (79) we use the bound from Theorem 1, and finally, in (80) GNN/MNN outputs are normalized $B_f = 1$.

7 Proof of Theorem 2

Theorem 2 bounds the convergence of a Graph Transformer with GNN-based PEs to a Manifold Transformer with MNN-based PEs.

Proof. The graph transformer's output for the i-th node can be written in vector form as

$$\mathbf{x}_i = \frac{\sum_{j=1}^n \exp[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle]\mathbf{V}\mathbf{x}_j}{\sum_{j=1}^n \exp[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle]} \tag{81}$$

We will introduce an auxiliary term built from the induced manifold signal of the sample output of a MT. For point $x \in \mathcal{M}$

$$\bar{\mathbf{\Phi}}_{\mathcal{M}}(f;\mathbf{T})(x) = \left(\frac{\mathbf{I}_N \mathbf{P}_N \int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y)\, d\mu(y)}{D_{\mathcal{M}}}\right)(x) = \frac{\sum_{j=1}^n \int_{V_j} \exp[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle]\mathbf{V}f(x_j)\, d\mu(y)}{\int_{\mathcal{M}} \exp\left[\langle \mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle\right]\, d\mu(y)}, \tag{82}$$

The output difference for node i can be decomposed as

$$\|\mathbf{\Phi}_G(\mathbf{X};\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\| \tag{83}$$

$$= \|\mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T})(x) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(f;\mathbf{T})(x) + \bar{\mathbf{\Phi}}_{\mathcal{M}}(f;\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\| \tag{84}$$

$$\leq \|\mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T})(x) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathbf{X};\mathbf{T})(x)\| + \|\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathbf{X};\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f,\mathcal{L};\mathbf{T})(x)\| \tag{85}$$

From Equation (83) to Equation (84) we add and subtract the induced manifold signal term, and in (84) to (85) we use the triangle inequality.

For notational brevity, we will denote the GT attention coefficients as $\gamma_{ij} = \exp[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle]$, the MT attention coefficients as $\tilde{\gamma_{iy}} = \exp[\langle \mathbf{Q}f(x_i)\mathbf{K}f(y) \rangle]$, and the induced manifold signal coefficients as $\tilde{\gamma}_{ij} = \exp[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j) \rangle]$. Furthermore, when necessary we will abbreviate the denominator terms as $D_{\mathbf{G}} = \sum_{j=1}^{N} \gamma_{ij}$ and $D_{\mathcal{M}} = \int_{\mathcal{M}} \tilde{\gamma}_{iy} d\mu(y)$.

Thus we have

$$\mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T})(x_i) = \frac{\sum_{j=1}^{n} \gamma_{ij} \mathbf{V}\mathbf{x}_j}{D_{\mathbf{G}}} \tag{86}$$

$$\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) = \frac{\int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V}f(y) d\mu(y)}{D_{\mathcal{M}}} \tag{87}$$

$$\bar{\mathbf{\Phi}}_{\mathcal{M}}(f;\mathbf{T})(x_i) = \frac{\sum_{j=1}^{n} \tilde{\gamma}_{ij} \mathbf{V}f(x_j)}{D_{\mathcal{M}}} \tag{88}$$

We will now bound the first term of (85),

$$\left| \mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T})(x) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathbf{X};\mathbf{T})(x) \right| = \left| \frac{\sum_{j=1}^{n} \gamma_{ij} \mathbf{V}\mathbf{x}_j}{D_{\mathbf{G}}} - \frac{\sum_{j=1}^{n} \tilde{\gamma}_{ij} \mathbf{V}f(x_j)}{D_{\mathcal{M}}} \right|. \tag{89}$$

Distribute the denominators, add a subtract $D_{\mathcal{M}}\gamma_{ij}\mathbf{V}f(x_j)$, then apply triangle inequality:

$$\left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \left[\sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}\mathbf{x}_j - D_{\mathbf{G}} \tilde{\gamma}_{ij} \mathbf{V}f(x_j) \right] \right| \tag{90}$$

$$= \left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \left[\sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}\mathbf{x}_j - D_{\mathcal{M}} \gamma_{ij} \mathbf{V}f(x_j) + D_{\mathcal{M}} \gamma_{ij} \mathbf{V}f(x_j) - D_{\mathbf{G}} \tilde{\gamma}_{ij} \mathbf{V}f(x_j) \right] \right| \tag{91}$$

$$\leq \left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}\mathbf{x}_j - D_{\mathcal{M}} \gamma_{ij} \mathbf{V}f(x_j) \right| + \left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}f(x_j) - D_{\mathbf{G}} \tilde{\gamma}_{ij} \mathbf{V}f(x_j) \right| \tag{92}$$

$$= \left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}(\mathbf{x}_j - f(x_j)) \right| + \left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \sum_{j=1}^{N} (D_{\mathcal{M}} \gamma_{ij} - D_{\mathbf{G}} \tilde{\gamma}_{ij}) \mathbf{V}f(x_j) \right|. \tag{93}$$

Note that $\mathbf{x}_j - f(x_j)$ corresponds to the output difference between a GNN and an MNN. This difference can be bounded as per Theorem 1, which denote as Δ_{GNN}. Thus, the first term of (93) can be bounded as

$$\left| \frac{1}{D_{\mathbf{G}} D_{\mathcal{M}}} \sum_{j=1}^{N} D_{\mathcal{M}} \gamma_{ij} \mathbf{V}(\mathbf{x}_j - f(x_j)) \right| \leq |V| \cdot \sum_{j=1}^{N} \gamma_{ij} \left| \mathbf{x}_j - f(x_j) \right| \tag{94}$$

$$\leq \frac{1}{D_{\mathbf{G}}} C_V \Delta_{GNN} \sum_{j=1}^{N} \gamma_{ij} = C_V \Delta_{GNN}, \tag{95}$$

by using triangle inequality and the bound on the linear operator $\mathbf{V}$, then the bound on Δ_{GNN} and the definition for $D_{\mathbf{G}}$.

Now we bound the second term in (93). We add and subtract another comparative term $D_{\mathcal{M}}\tilde{\gamma}_{ij}$,

$$\left|\frac{1}{D_{\mathbf{G}}D_{\mathcal{M}}}\sum_{j=1}^{N}(D_{\mathcal{M}}\gamma_{ij}-D_{\mathbf{G}}\tilde{\gamma}_{ij})\mathbf{V}f(x_j)\right| \tag{96}$$

$$=\left|\frac{1}{D_{\mathbf{G}}D_{\mathcal{M}}}\sum_{j=1}^{N}\Big[(D_{\mathcal{M}}\gamma_{ij}-D_{\mathcal{M}}\tilde{\gamma}_{ij})+(D_{\mathcal{M}}\tilde{\gamma}_{ij}-D_{\mathbf{G}}\tilde{\gamma}_{ij})\Big]\mathbf{V}f(x_j)\right| \tag{97}$$

$$=\left|\frac{1}{D_{\mathbf{G}}D_{\mathcal{M}}}\sum_{j=1}^{N}[(D_{\mathcal{M}}(\gamma_{ij}-\tilde{\gamma}_{ij})+(D_{\mathcal{M}}-D_{\mathbf{G}})\tilde{\gamma}_{ij}]\mathbf{V}f(x_j)\right| \tag{98}$$

$$\leq\sum_{j=1}^{N}\left|\frac{D_{\mathcal{M}}}{D_{\mathbf{G}}D_{\mathcal{M}}}\left[((\gamma_{ij}-\tilde{\gamma}_{ij})\right]\mathbf{V}f(x_j)\right|+\left|\frac{\tilde{\gamma}_{ij}(D_{\mathcal{M}}-D_{\mathbf{G}})}{D_{\mathcal{M}}D_{\mathbf{G}}}\mathbf{V}f(x_j)\right| \tag{99}$$

$$\leq\sum_{j=1}^{N}\left[\frac{1}{D_{\mathbf{G}}}|\gamma_{ij}-\tilde{\gamma}_{ij}|+\frac{\tilde{\gamma}_{ij}}{D_{\mathcal{M}}D_{\mathbf{G}}}|D_{\mathcal{M}}-D_{\mathbf{G}}|\right]C_VB_f \tag{100}$$

On (98) we rearrange the terms, (99) we apply triangle inequality, on (100) the linear operator bounds using Assumption 2, bound on the MNN signal B_f using Assumption 3. Applying the bound of Lemma 6 (80) to (100) we obtain

$$\frac{N\cdot 2e^MC_{QK}B_f}{D_{\mathbf{G}}}+\sum_{j=1}^{N}\frac{\tilde{\gamma}_{ij}C_VB_f|D_{\mathcal{M}}-D_{\mathbf{G}}|}{D_{\mathcal{M}}D_{\mathbf{G}}}\leq 2e^MC_{QK}\Delta_{\text{GNN}}+|D_{\mathcal{M}}-D_{\mathbf{G}}| \tag{101}$$

We now turn to bound the term $|D_{\mathcal{M}}-D_{\mathbf{G}}|$.

$$|D_{\mathcal{M}}-D_{\mathbf{G}}|=\left|\int_{\mathcal{M}}\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(y)\rangle\right]\,d\mu(y)-\sum_{j=1}^{N}\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(x_j)\rangle\right]\right| \tag{102}$$

The partitions $B_r(x_i)\subset V_i$ cover the manifold $\mathcal{M}$. Therefore, for every $x\in\mathcal{M}$, there exists a x_i such that $|x-x_i|\leq r$. We can decompose the integral as

$$\left|\sum_{i=1}^{N}\int_{V_j}\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(y)\rangle\right]d\mu(y)-\sum_{j=1}^{N}\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(x_j)\rangle\right]\right| \tag{103}$$

$$\leq\left|\sum_{i=1}^{N}\int_{V_j}\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(y)\rangle\right]-\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(x_j)\rangle\right]d\mu(y)\right| \tag{104}$$

$$\leq\sum_{i=1}^{N}\int_{V_j}\left|\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(y)\rangle\right]-\exp\left[\langle\mathbf{Q}f(x_i),\mathbf{K}f(x_j)\rangle\right]\right|d\mu(y) \tag{105}$$

$$\leq e^MC_{QK}B_f\|f(x_j)-f(y)\|\leq C_{QK}B_fr=e^MC_{QK}r. \tag{106}$$

In Equation (104), we add repeated negative terms in order to bring the partition center coefficients into the integral, in (105) the triangle inequality, in (106) the normalized Lipschitz property of the MNN output (Assumption 4) and the bound $|x_j-y|\leq r$ for all $y\in V_j$. Finally we set $B_f=1$.

Combining (95), (106) and (101), we conclude the bound for the first term of (85) is

$$\left|\mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T})(x)-\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathbf{X};\mathbf{T})(x)\right|\leq C_V\Delta_{GNN}+2e^MC_{QK}\Delta_{GNN}+e^MC_{QK}r \tag{107}$$

$$=(C_V+2e^MC_{QK})\Delta_{GNN}+e^MC_{QK}r \tag{108}$$

We now bound the second term of (85),

$$\|\hat{\mathbf{\Phi}}_{\mathcal{M}}(\mathbf{X};\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f,\mathcal{L};\mathbf{T})(x_i)\| \tag{109}$$

$$= \left\| \frac{\sum_{j=1}^{n} \tilde{\gamma}_{ij} \mathbf{V} f(x_j)}{\int_{\mathcal{M}} \tilde{\gamma}_{iy} d\mu(y)} - \frac{\int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y)}{\int_{\mathcal{M}} \tilde{\gamma}_{iy} d\mu(y)} \right\| \tag{110}$$

$$\leq \left\| \frac{1}{D_{\mathcal{M}}} \sum_{j=1}^{n} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y) - \int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \right\| \tag{111}$$

$$= \left\| \frac{1}{D_{\mathcal{M}}} \sum_{j=1}^{n} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \right\| \tag{112}$$

$$\leq \frac{1}{D_{\mathcal{M}}} \sum_{j=1}^{n} \int_{V_j} \|\tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{\gamma}_{iy} \mathbf{V} f(y)\| \, d\mu(y) \tag{113}$$

In Equation (111) we apply the definition of $D_{\mathcal{M}}$ and repeat each term for each element in the partition. In Equation (112) we rearrange the integral, and in (113) we apply the triangle inequality.

The integrand can be bounded as

$$\|\tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{\gamma}_{iy} \mathbf{V} f(y)\| \tag{114}$$

$$= \|\tilde{\gamma}_{ij} \left[\mathbf{V} f(x_j) - \mathbf{V} f(y)\right] - \left[\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}\right] \mathbf{V} f(y)\| \tag{115}$$

$$\leq |\tilde{\gamma}_{ij}| \, \|\mathbf{V} f(x_j) - \mathbf{V} f(y)\| - |\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| \, \|\mathbf{V} f(y)\| \tag{116}$$

$$\leq |\tilde{\gamma}_{ij}| C_V r - C_V |\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| \tag{117}$$

$$\leq |\tilde{\gamma}_{ij}| C_V r - e^M C_{QKV} r \tag{118}$$

where in (115) we add and subtract $\tilde{\gamma}_{ij} \mathbf{V} f(y)$, in (116) apply the triangle inequality, in (117) we use the linear operator bounds, and $\|x_j - y\| \leq r$, and the assumption that the MNN output normalized Lipschitz. Finally in (118) we apply the bound of Lemma 5 with $B_f = 1$.

Applying (118) into (113), we obtain that the second term is bounded by

$$\frac{1}{D_{\mathcal{M}}} \sum_{j=1}^{n} \int_{V_j} |\tilde{\gamma}_{ij}| C_V r - e^M C_{QKV} r \, d\mu(y) \leq C_V r \tag{119}$$

using the fact that $\sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} d\mu(y) \leq D_{\mathcal{M}}$ and that the second term is dominated by $D_{\mathcal{M}}$.

Putting together Equations (108) and (119) , we have that

$$\|\mathbf{\Phi}_{\mathbf{G}}(\mathbf{X};\mathbf{T}) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})\| \leq (C_V + 2e^M C_{QK}) \Delta_{GNN} \tag{120}$$

$$+ e^M C_{QK} r \tag{121}$$

$$+ C_V r \tag{122}$$

Grouping the terms that depend on r gives us the statement of Theorem 2. □

8 Proof of Corollary 3

Corollary 3 bounds the output difference between two GT's trained with differently sized graphs by applying Theorem 2.

Proof. The output difference can be bounded as

$$\frac{1}{\mu(\mathcal{M})}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T}) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathbf{X}_2;\mathbf{T})\|_{L^{1,2}(\mathcal{M})} \tag{123}$$

$$= \frac{1}{\mu(\mathcal{M})}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T}) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T}) + \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T}) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathbf{X}_2;\mathbf{T})\|_{L^{1,2}(\mathcal{M})} \tag{124}$$

$$\leq \frac{1}{\mu(\mathcal{M})}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T}) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})\|_{L^{1,2}(\mathcal{M})} \tag{125}$$

$$+ \frac{1}{\mu(\mathcal{M})}\|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T}) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathbf{X}_2;\mathbf{T})\|_{L^{1,2}(\mathcal{M})} \tag{126}$$

$$\leq \frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}} \|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{127}$$

$$+ \frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}} \|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathbf{X}_2;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{128}$$

where in (124) we add and subtract $\mathbf{\Phi}_{\mathcal{M}}(f;,\mathbf{T})$, in (125) we apply the triangle inequality, and (127) use the definition of $L^{1,2}(\mathcal{M})$.

The two terms in (128) correspond to the pointwise difference between the induced manifold signal of GT and the output signal of the MT, for $x \in \mathcal{M}$. Consider bounding the first term, that compares $\mathbf{\Phi}_{\mathbf{G}_1}$ with $\mathbf{\Phi}_{\mathcal{M}}$,

$$\frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}} \|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{129}$$

$$= \frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N}\int_{V_i} \|\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{130}$$

$$= \frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N}\int_{V_i} \|\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) \tag{131}$$

$$+ \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{132}$$

$$\leq \frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N}\int_{V_i} \|\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i)\| \tag{133}$$

$$+ \|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \, d\mu(x) \tag{134}$$

where we add and subtract the MNN output at point x_i, and apply triangular inequality.

The first term of (138) is the statement of Theorem 2. The second term is the output difference of MT between a point within a partition $x \in V_i$ and the center of the ball containing the partition x_i, therefore it holds that $\|x_i - x\| \leq r$. Denote the softmax denominators over x and x_i as $D_{\mathcal{M}}(x)$ and $D_{\mathcal{M}}(x_i)$ respectively. We can decompose this as

$$\|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \tag{135}$$

$$\leq \left\| \frac{\int_{\mathcal{M}} \tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y)}{\int_{\mathcal{M}} \tilde{\gamma}_{iy}d\mu(y)} - \frac{\int_{\mathcal{M}} \tilde{\gamma}_{xy}\mathbf{V}f(y)d\mu(y)}{\int_{\mathcal{M}} \tilde{\gamma}_{xy}d\mu(y)} \right\| \tag{136}$$

$$\leq \left\| \frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)d\mu(y) \right\| \tag{137}$$

$$\leq \frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} \|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\| \; d\mu(y) \tag{138}$$

Which we obtain by distributing the denominators, applying the triangle inequality, and grouping terms. The integrand in (138) is bounded as

$$\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\| \tag{139}$$
$$\leq \|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\left[\mathbf{V}f(x) - \mathbf{V}f(y)\right] - \left[D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\right]\mathbf{V}f(y)\| \tag{140}$$
$$\leq D_{\mathcal{M}}(x)C_V\tilde{\gamma}_{iy}\|f(x) - f(y)\| - C_V B_f|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}| \tag{141}$$

We now focus on the second term of (141),

$$\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\| \leq D_{\mathcal{M}}(x)|\tilde{\gamma}_{iy} - \tilde{\gamma}_{xy}| + \|\left[D_{\mathcal{M}}(x) - D_{\mathcal{M}}(x_i)\right]\tilde{\gamma}_{xy}\| \tag{142}$$

Further decopose the first term of (142),

$$|\tilde{\gamma}_{iy} - \tilde{\gamma}_{xy}| \tag{143}$$
$$= |\exp\langle\mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle - \exp\langle\mathbf{Q}f(x), \mathbf{K}f(y)\rangle| \tag{144}$$
$$= |\exp\langle\mathbf{Q}\left[f(x_i) - f(x)\right], \mathbf{K}f(y)\rangle| \tag{145}$$
$$\leq L_E L_{\text{MNN}} C_{QK} r. \tag{146}$$

The second term of (142) is

$$\tilde{\gamma}_{xy}\|D_{\mathcal{M}}(x) - D_{\mathcal{M}}(x_i)\| \tag{147}$$
$$= \left\|\int_{\mathcal{M}} \tilde{\gamma}_{xy}\tilde{\gamma}_{xy} - \tilde{\gamma}_{iy}d\mu(y)\right\| \tag{148}$$
$$\leq \int_{\mathcal{M}} \tilde{\gamma}_{xy}\left\|\tilde{\gamma}_{xy} - \tilde{\gamma}_{iy}\right\| d\mu(y) \tag{149}$$
$$\leq D_{\mathcal{M}}(x_i)L_E L_{\text{MNN}} C_{QK} r \tag{150}$$

where we again we use the bound in (146) to upper bound the remaining integral by $D_{\mathcal{M}}(x_i)$. Using Equations (150) and (146) we finish the integrand bound in (142). Returning to the integral (138), we have

$$\frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} \|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\|\ d\mu(y) \tag{151}$$
$$\leq \frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} D_{\mathcal{M}}(x)C_V L_{\text{MNN}}|x-y| - C_V B_f D_{\mathcal{M}}(x_i)L_E L_{\text{MNN}} C_{QK} r\ d\mu(y) \tag{152}$$
$$\leq C_V B_f C_{QK} r \tag{153}$$

From (151) to (152) we use the fact that the output difference in the first term of the integrand is dominated by the exponentials in $\int_{\mathcal{M}} \gamma_{iy}d\mu(y)$, concluding that it vanishes. The second term, we can bound by $D_{\mathcal{M}}(x)$ to cancel with the denominator term. We conclude that

$$\|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2 \leq C_V C_{QK} r \tag{154}$$

We can now bound (127) by applying this bound and the bound of Theorem 2,

$$\frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}} \|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x) - \mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2\ d\mu(x) \leq \Delta_{\mathbf{\Phi}(\mathbf{X}_1)} + C_V C_{QK} r, \tag{155}$$

where $\Delta_{\mathbf{\Phi}(\mathbf{X}_1)}$ denotes the bound of Theorem 2 for G_1.

Applying this bound in (128) for $\mathbf{\Phi}(\mathbf{X}_1)$ and $\mathbf{\Phi}(\mathbf{X}_2)$,

$$\frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N}\int_{V_i}\|\mathbf{\Phi}_{\mathbf{G}_1}(\mathbf{X}_1;\mathbf{T})(x_i)-\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i)\| \tag{156}$$

$$+\|\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x_i)-\mathbf{\Phi}_{\mathcal{M}}(f;\mathbf{T})(x)\|_2\,d\mu(x) \tag{157}$$

$$\leq \Delta_{\mathbf{\Phi}(\mathbf{X}_1)}+\Delta_{\mathbf{\Phi}(\mathbf{X}_2)}+2C_V C_{QK} r, \tag{158}$$

gives us the statement of Corollary 3. □

9 Experiment Implementation Details

Datasets. Snap-patents is a network for patents granted between 1963 to 1999 in the US. Each node is a patent, and a directed edge connects a patent to another patent that it cited. The prediction task is to classify each patent into one of five time intervals. ArXiv-year is a paper citation network on the arXiv papers. Each node represents a paper, and a directed edge connects a paper to another paper that it cited. The task is to predict the posting time of each paper, which is classified into one of five time intervals between 2013 and 2020. Both snap-patents and arXiv-year are heterophilic graph datasets. OGBN-MAG is a heterogeneous network composed of a subset of the Microsoft Academic Graph (MAG), capturing the relationships among papers, authors, institutions and topics. The node classification task is to predict the venue of each paper. REDDIT-BINARY consists of 2000 graphs, each representing a subreddit community. The graph-level binary classification task is to identify each community as either a Q&A community or discussion-based community using graph structures.

Table 1: Datasets used for transferability experiments

Dataset	Nodes	Edges	Max Train/Test Nodes	Classes	Graphs	Feature Dim
snap-patents	2,923,922	13,975,788	334,000	5	1	269
arXiv-year	169,343	1,166,243	90,000	5	1	128
OGBN-MAG	736,389	5,416,271	90,000	349	1	128
REDDIT-BINARY	avg. 429.61	avg. 497.75	/	2	2000	0

Dataset preparation. Each dataset is split into train/val/test fractions of 50%-25%-25% respectively. For datasets that have pre-established train/test masks, we discard the masks in favor of our partition proportions. The training and testing partitions are the sources for the graph subsampling procedure explained in Section 4.

Training procedure and transferability evaluation. For each model architecture, we train multiple models with graphs of increasing sizes and evaluate each model on testing graphs of different sizes. For single-graph datasets, the training graphs and testing graphs are constructed by subsampling a fraction of nodes from the training split and the testing split respectively. For multi-graph datasets, we construct the training graphs and testing graphs by subsampling the nodes in each training graph and each testing graph respectively by a specific fraction. We do not create graph batches, but rather train with the full subsampled graph.

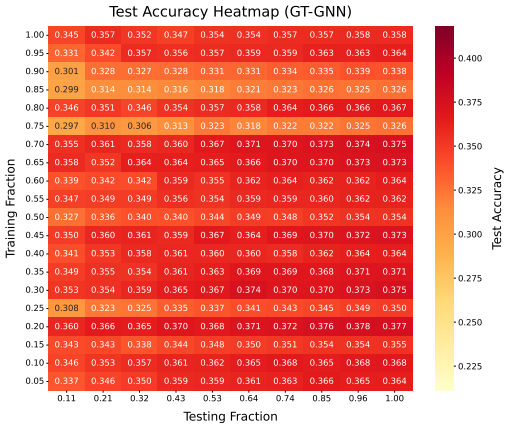
Hyperparameters. The hyperparameters used for each model and dataset are available in Table 2.

10 Extended Results

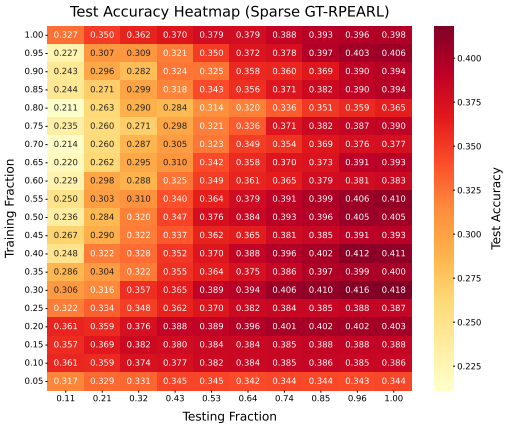
In this section, we provide additional results for MAG and Reddit datasets, as well as the full heatmaps for every model-dataset combination. In MAG and Reddit GNN and GTs show comparable performance and transferability patterns. As the training fraction increases, the total number of nodes decreases. In the cases of SNAP-patents and MAG, Exphormers presents a monotonically increasing performance as training graph size increases.

Table 2: Hyperparameters for different model architectures across datasets

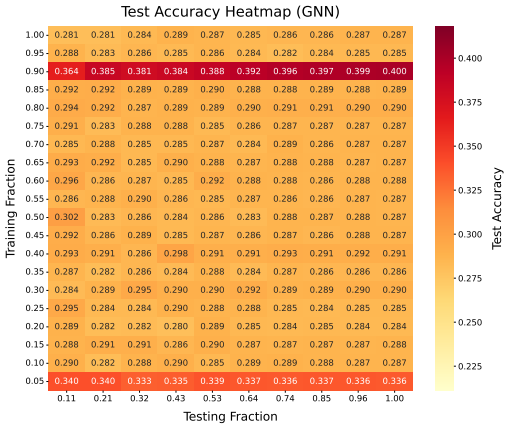
Model/Hyperparameters	snap-patents	arXiv-year	OGBN-MAG	REDDIT-BINARY
General				
Batch size	32	32	32	16
Max epochs	300	300	700	500
Pooling	-	-	-	sum
GNN				
Learning rate	0.01	0.01	0.01	7×10^{-3}
Dropout	0.5	0.5	0.5	0
Hidden channels	256	256	256	512
Number of layers	3	3	4	4
GT				
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Transformer dropout	0.25	0.25	0.25	0
Transformer Heads	4	4	4	4
Transformer dim feedforward	128	128	128	512
Transformer d model	128	128	128	64
Transformer number of layers	3	3	3	6
PE embedding	GNN	GNN	GNN	GNN/RPEARL
PE dropout	0.025	0.15	0.15	0
PE hidden channels	128	128	128	512
PE number of layers	8	8	8	3
Sparse GT				
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-5}
Dropout	0.05	$\frac{N}{1+12N}$	0.5	0.01
D model	128	128	128	128
Heads	8	8	8	8
Number of hops	2	1	2	2
Number of layers	3	3	3	3
PE embedding	RPEARL	RPEARL	RPEARL	RPEARL
Exphormer				
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}	-
Dropout	0.5	0.5	0.5	
D model	256	256	256	
Dim feedforward	512	512	512	
Expander algorithm	Random-d	Random-d	Random-d	
Expander degree	3	3	3	
Heads	8	8	8	
Number of layers	2	2	2	



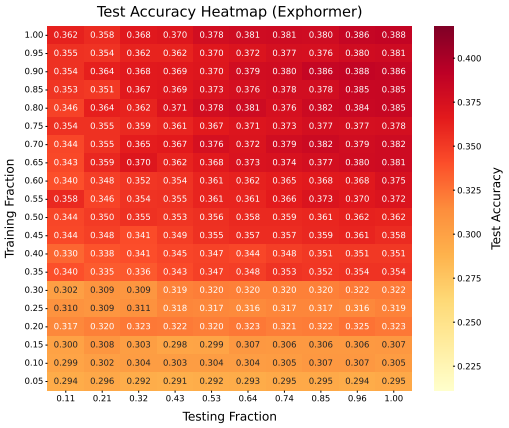
(a) GT-GNN



(b) Sparse GT-RPEARL

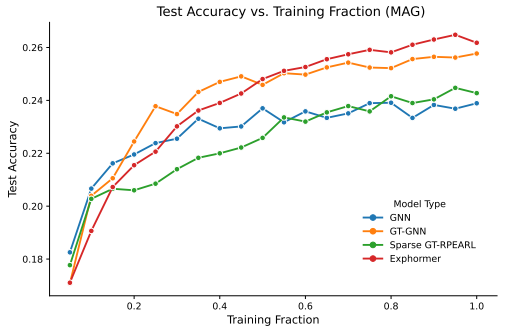


(c) GNN

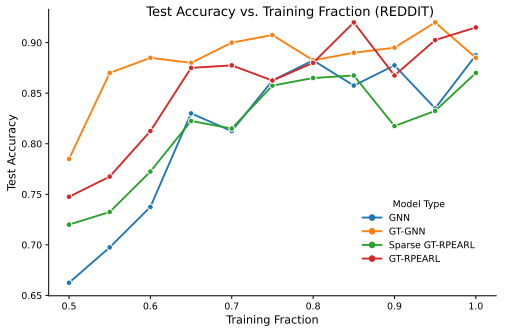


(d) Exphormer

Figure 4: Test Accuracy Heatmaps on arXiv-year across Models

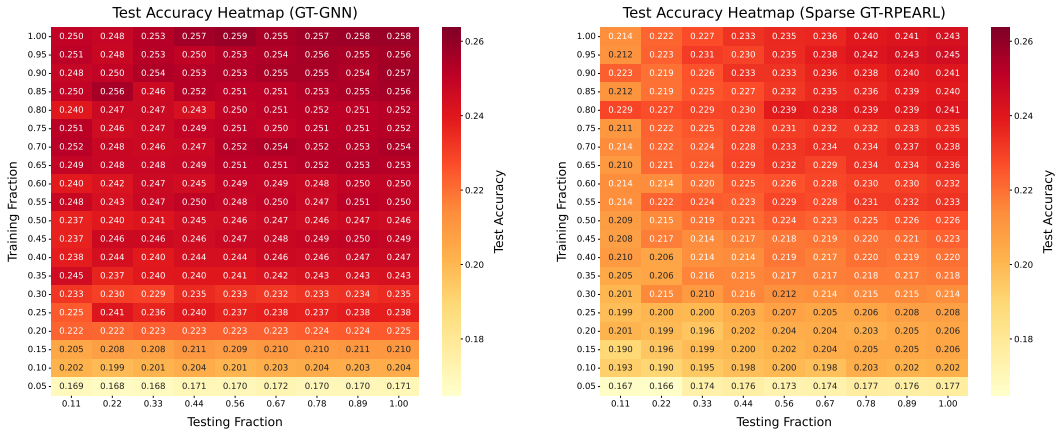


(a) OGBN-MAG



(b) REDDIT-BINARY

Figure 5: Transferability results for OGBN-MAG and REDDIT-BINARY.



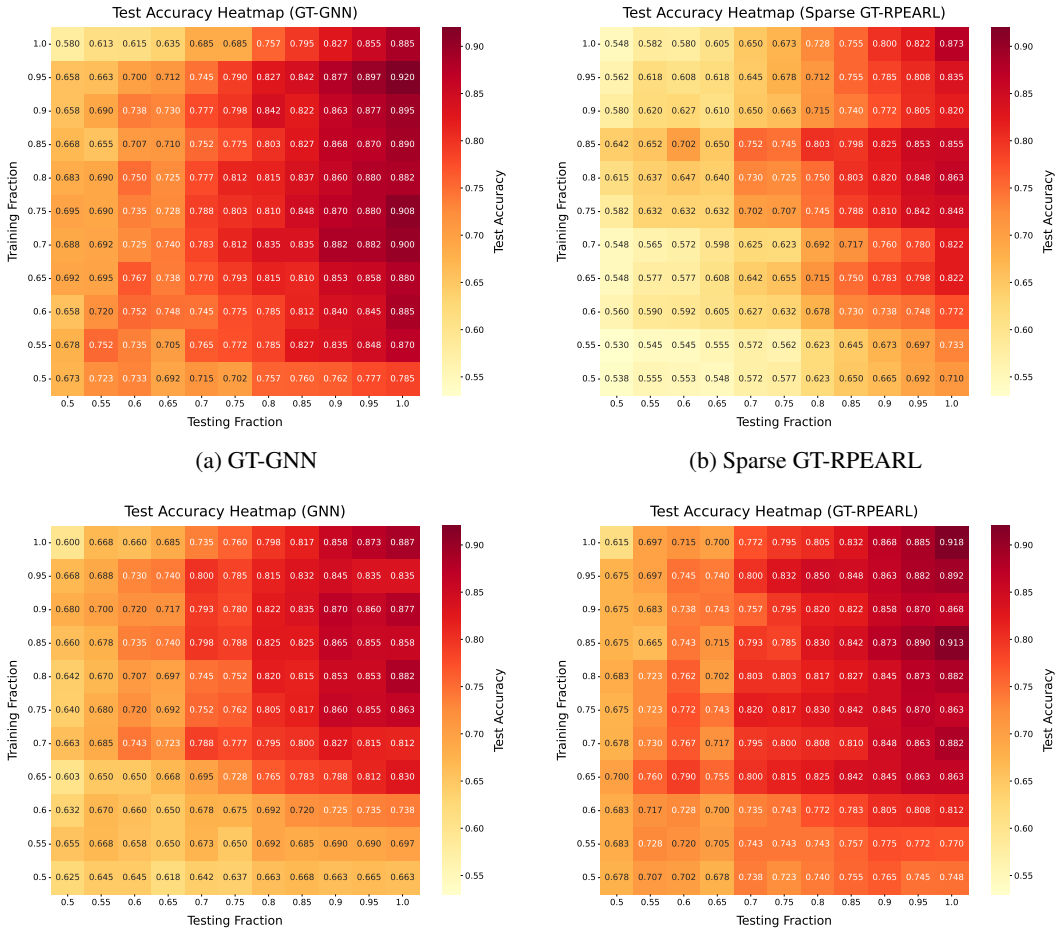
(a) GT-GNN

(b) Sparse GT-RPEARL

(c) GNN

(d) Exphormer

Figure 6: Test Accuracy Heatmaps on OGBN-MAG across Models



(a) GT-GNN

(b) Sparse GT-RPEARL

(c) GNN

(d) GT-RPEARL

Figure 7: Test Accuracy Heatmaps on REDDIT-BINARY across Models