SIZE-TRANSFERABILITY OF GRAPH TRANSFORMERS WITH CONVOLUTIONAL POSITIONAL ENCODINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention-based architectures have achieved remarkable success across data modalities, motivating the rise of Graph Transformers (GTs). GTs typically encode structural information via positional encodings (PEs), which may be either embedding vectors added to the input or modifications to the attention kernel. Given the challenges of learning from large graphs, a crucial question is whether GTs are *size-transferable*, that is, whether they can generalize to graphs larger than those seen during training. In this work, we study GTs through the lens of manifold limit models and establish that GTs can *inherit* the transferability properties of their PEs. In particular, if the PEs are transferable, the GT is also transferable. We derive bounds showing that GTs with Graph Neural Network (GNN)-based PEs are transferable. Furthermore, our results show that their transferability is more stable than that of eigenvector-based GTs. We complement our theory with extensive experiments on standard graph benchmarks, where we confirm that GTs with GNN-based PEs exhibit superior transferability compared to those with eigenvector-based PEs. We showcase a real-world application of a computationally efficient GT in a shortest-path distance estimation task over terrain manifolds. Our results provide new insights into GTs and suggest practical directions for efficient training in large-scale settings.

1 INTRODUCTION

Transformers have been adapted to graph-structured data by injecting adjacency information into node features via positional encodings (PEs) followed by attention, yielding Graph Transformers (GTs). They have become highly competitive across several domains, including large-scale molecular property prediction (Ying et al., 2021), biomedical knowledge graphs (Hu et al., 2020), and long-range data benchmarks (Dwivedi et al., 2022). The key feature of GTs is performing pairwise attention between node features beyond their local connectivity, which improves their capacity to process long-range interactions.

However, this comes at the cost of increased computational complexity. Full pairwise attention over graphs scales quadratically with the number of nodes, which greatly increases the amount of data and computation required for training. Since this makes training directly on large-scale graphs prohibitively expensive, it is paramount to develop GTs that generalize to larger graphs than those seen during training, a property known as *size-transferability*.

Theoretical foundations pertaining to GT size-transferability are largely unexplored. In contrast, this key property is well-established for Graph Neural Networks (GNNs), which are composed of graph convolutional filters. Extensive theory from spectral graph signal processing establishes that graph filters and GNNs—under continuity conditions—are stable to perturbations and transferable across graphs sampled from the same limit model (Wang et al., 2024a;b; Ruiz et al., 2023; 2020; Keriven et al., 2020). These results are possible because graph convolutions incorporate graph structure by construction.

A central insight across the GT literature is that encoding graph structure explicitly using PEs enables effective attention-based architectures. Examples of PEs are embedding vectors added to node features, such as the top-K eigenvectors of the graph Laplacian (Dwivedi & Bresson, 2020), which generalize the sinusoidal encodings of conventional Transformers. However, the transferability of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

eigenvector encodings depends on the spectral gap of the limit operator at the truncation point (Dunson et al., 2021), a quantity that is not known in advance.

In this work, we present theoretical results that link the transferability of GTs to the transferability of their PEs. In particular, our main theoretical contribution is to show that GTs *inherit* size-transferability from their PEs. We state this in our main theoretical contribution:

[C1] We provide the first theoretical guarantees of GT transferability in Section 4. Namely, we show that GTs converge to a Manifold Transformer, their limiting object in the manifold domain (Theorem 1). Leveraging this result, we show that if the PEs are transferable, size-transferability holds with a rate that depends on the encoding's transferability bound (Corollary 2). Our theory extends readily to other Transformer variants.

Our results indicate that the choice of PE is a critical ingredient for deploying GTs across graphs of different sizes. This leads us to contrast the transferability properties of two different positional encodings through the lens of spectral theory and established results on GNN size transferability:

[C2] We compare the convergence of two popular encodings to their manifold limits: Laplacian eigenvector PEs and GNN-based PEs. We show how eigenvector PE convergence depends on the spectral gap (Theorem 3), which is unobservable in practice. In contrast, the convergence of GNN-based PEs depends on Lipschitz constants, which are controllable.

Finally, we empirically verify the transferability properties of the architectures in two settings.

[C3] In Section 6, we carry out experiments that illustrate transferable performance of GTs with eigenvector PEs and a GNN-based PE (RPEARL) on citation and social network datasets (arXiv-year, Reddit, SNAP-Patents, MAG). We empirically confirm that neighborhood-masked GTs, used for computational efficiency, also exhibit transferable performance. An extensive empirical verification of other popular GT variants and GNN PE backbones is provided in Appendix G.4.

[C4] As an application, we consider shortest path distance estimation over terrain graphs, using a transferable and computationally efficient GT.

2 Related Work

Limit models provide a foundation for analyzing GNN transferability, stability, and statistical generalization, including manifolds (Wang et al., 2024a;b; 2025), graphons (Ruiz et al., 2020; Maskey et al., 2025; 2023), and general limit objects (Levin et al., 2025). Transformer length generalization has been studied empirically and theoretically (Zhou et al., 2024; Anil et al., 2022; Huang et al., 2024a), with positional encodings (PEs) playing an important role (Kazemnejad et al., 2023; Peng et al., 2023). The study of GTs has been focused on positional encodings, including absolute spectral or learned PEs (Kreuzer et al., 2021; Cantürk et al., 2024; Kanatsoulis et al., 2024), relative distance or diffusion encodings (Mialon et al., 2021; Ying et al., 2021; Ma et al., 2023), and sparse attention masks (Shirzad et al., 2023). Related analyses address PE invariance, stability, and expressivity (Lim et al., 2023; Huang et al., 2024b; Black et al., 2024; Stoll et al., 2025). We compliment these studies by addressing a limit model framework to analyze the size transferability of graph transformers with an explicit connection to the properties of positional encodings. In Section A in the appendix, we discuss more related work.

3 Graph Transformers with Convolutional Positional Encodings

Preliminary notation. 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 $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix and the diagonal matrix $\mathbf{D}$ is called the degree matrix. Node signals are vectors denoted as $\mathbf{x}_i \in \mathbb{R}^q$ for each node $i \in [1, N]$. We collect node signals into a graph signal matrix of column vectors $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$.

A GT is a layered architecture that processes a graph signal $\mathbf{X}$, and outputs another signal $\mathbf{Y} \in \mathbb{R}^{q \times N}$ via an attention (Vaswani et al., 2017) operation, where each layer computes

$$\mathbf{Y} = \mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X}) = \mathbf{V}\mathbf{X} \text{ softmax}\left[(\mathbf{Q}\mathbf{X})^{\top}(\mathbf{K}\mathbf{X})\right]^{\top}. \tag{1}$$

The network $\mathbf{\Phi}_{\mathbf{G}}$ is parameterized by the query, key and value linear maps $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{q \times q}$, collected in $\mathcal{T} = \{\mathbf{Q}, \mathbf{K}, \mathbf{V}\}$. The operation $(\mathbf{Q}\mathbf{X})^{\top}(\mathbf{K}\mathbf{X})$ computes inner products between pairs of projected node features $\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle$, for all $i, j \in [1, N]$. Then, the softmax operation normalizes the pairwise inner products into probability vectors along each row. Finally, the output node features are computed as a linear combination of the projected values weighted by the attention coefficients.[1]

Structural information is introduced by addition to the input $\mathbf{X}_0 = \mathbf{X}_{\text{in}} + \mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}, \mathbf{Z})$, where $\mathbf{\Psi}_{\mathbf{G}} : \mathbb{R}^{p \times N} \to \mathbb{R}^{q \times N}$ is called the graph *positional encoder* and $\mathbf{X}_{\text{in}}$ is the graph input signal. The operator $\mathbf{\Psi}_{\mathbf{G}}$ maps each node into a positional encoding vector utilizing the structural information of $\mathbf{L}$, and another graph signal $\mathbf{Z}$. Given that $\mathbf{\Psi}_{\mathbf{G}}$ is the only piece of the architecture that depends on the graph structure, the choice of $\mathbf{\Psi}_{\mathbf{G}}$ is crucial for effective graph learning. Popular examples include top-k eigenvectors of $\mathbf{L}$ (Dwivedi & Bresson, 2020), or learnable embeddings based on powers of the random walk matrix (Dwivedi et al., 2021).

3.1 Positional encodings with Graph Neural Networks

Consider a GT with a graph neural network (GNN) acting as a learnable positional encoder $\mathbf{\Psi}_{\mathbf{G}}$. A GNN is a layered architecture composed of graph convolutional filters followed by pointwise nonlinearities. One layer of a GNN is written as

$$\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}, \mathbf{Z}) = \sigma\left[\sum_{k=0}^{K-1} \mathbf{H}_k \mathbf{Z} \mathbf{L}^k\right] = \sigma\Big[h(\mathbf{L})\mathbf{Z}\Big] \tag{2}$$

where $\mathbf{Z} \in \mathbb{R}^{q \times N}$ stands for the input signals, $\mathcal{H} = \{\mathbf{H}_0, \ldots, \mathbf{H}_{K-1}\}$ collects the filter coefficients, $h(\mathbf{L}) : \mathbb{R}^{p \times N} \to \mathbb{R}^{q \times N}$ denotes the graph filter in linear operator form. Equation (2) performs a graph convolution of order K on the graph signals by the graph Laplacian $\mathbf{L}$, followed by the pointwise nonlinearity σ.

Remark 1. *Since GNNs are implicitly nonlinear functions of Laplacian eigenvectors (see Proposition 3.1 (Kanatsoulis et al., 2024)), they can be seen as an expressive alternative of Laplacian eigenvector-based positional encodings.*

4 Transferability Analysis of GTs via a Manifold Perspective

Transferability offers a principled way to overcome the practical limitations of quadratic attention by enabling models to capture structural regularities that persist across graph sizes and instances. In particular, if a sequence of graphs admits a well-defined limit object, then a model trained on small-scale graphs can, in principle, be transferred to larger-scale graphs without retraining, while preserving its functional behavior. In this section, we show that the transferability of GTs is inherited from the transferability of their PEs. That is, once PEs are size-transferable functions over the graph limit, the induced GT naturally generalizes across graph scales. Building on this insight, we present the first transferability analysis of GTs from a manifold-based perspective.

4.1 Discrete graphs and operator limits

We study transferability by relating graph-based computations to their continuum counterparts on a manifold. Let $\mathcal{M}$ be a d-dimensional compact, smooth and differentiable Riemannian submanifold 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

[1] For simplicity of presentation, we omit the transformer's feedforward layer, softmax normalization constant, and low-rank projections. We present our theory on a single-layer, single-head transformer, but its extension to these settings is straightforward.

$x \in \mathcal{M}$. The manifold data supported on each point $x \in \mathcal{M}$ is defined by vector-valued functions $g : \mathcal{M} \to \mathbb{R}^p$ (Wang et al., 2024b). We use $L^2(\mu; \mathbb{R}^p)$ to denote square-integrable $\mathbb{R}^p$-valued functions over $\mathcal{M}$ with respect to measure μ.

Given a set of N i.i.d. samples $X_N = \{x_i\}_{i=1}^N$ drawn from μ over $\mathcal{M}$, we construct a graph $\mathbf{G}_N = \mathbf{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ on these N sampled points X_N, where each point x_i is a vertex of graph $\mathbf{G}_N$, 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_N}, where ε_N is the kernel bandwidth. The kernel function is decided by the Euclidean distance $\|x_i - x_j\|$. The graph Laplacian denoted as $\mathbf{L}_N$ can be calculated based on the weight function (Merris, 1995). The constructed graph Laplacian with an appropriate kernel function has been proved to approximate the Laplace-Beltrami operator $\mathcal{L}$ of $\mathcal{M}$ (Calder & Trillos, 2022; Belkin & Niyogi, 2008; Dunson et al., 2021). In our theory, we implement the normalized Gaussian kernel definition in (Dunson et al., 2021).

To compare discrete and continuous computations over the sampled graphs and over the manifold, we introduce two linear mappings that connect data from sampled graph $\mathbf{G}_N$ and manifold $\mathcal{M}$ and back, i.e. a sampling operator as $\mathbf{S}_N : L^2(\mu; \mathbb{R}^p) \to \mathbb{R}^{p \times N}$, and an interpolation operator as $\mathbf{I}_N : \mathbb{R}^{p \times N} \to L^2(\mu; \mathbb{R}^p)$. We assume that these operators are bounded and consistent (Wang et al., 2024a; Levie et al., 2021), i.e.

$$\lim_{N \to \infty} \|\mathbf{I}_N \mathbf{S}_N g - g\|_{L^2(\mu)} = 0, \text{for all } g \in L^2(\mu; \mathbb{R}^p). \tag{3}$$

With these mapping operators, the convergence of graph Laplacians $\mathbf{L}_N$ over constructed graphs $\mathbf{G}_N$ to the continuous operator $\mathcal{L}$ can be written as

$$\lim_{N \to \infty} \|\mathbf{I}_N \mathbf{L}_N \mathbf{S}_N g - \mathcal{L} g\|_{L^2(\mu)} = 0, \text{for all } g \in L^2(\mu; \mathbb{R}^p). \tag{4}$$

4.2 Transferable Functions over Manifolds

We can now define transferability as a property of functions defined through graph operators that remain well-defined in the continuum limit. Let $\mathbf{\Psi}$ be a map defined on any Hilbert space with a bounded linear operator. We write this dependence explicitly as $\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g) \in L^2(\mu; \mathbb{R}^q)$ with $g \in L^2(\mu; \mathbb{R}^p)$ as a manifold-level realization, and $\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g) \in \mathbb{R}^{q \times N}$ with $\mathbf{S}_N g \in \mathbb{R}^{p \times N}$ as a graph-level realization. The same functional form $\mathbf{\Psi}$ is thus instantiated on discrete graphs and on the continuum, with dependence on the underlying domain entering solely through the operator $\mathbf{L}_N$ or $\mathcal{L}$. Both cases share the parametrization $\mathcal{H}$.

Definition 1 (Transferable functions over Manifolds)**.** *Assume that $\mathbf{\Psi}$ is permutation equivariant and uniformly Lipschitz in its inputs. $\mathbf{\Psi}$ is said to be transferable with respect to the operator convergence $\mathbf{L}_N \to \mathcal{L}$ if, for every admissible input function $g \in L^2(\mu; \mathbb{R}^p)$, for every $x_i \in X_N$ in samples of $\mathcal{M}$, it holds with probability at least $1 - \delta_N$,*

$$\left\| \mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)(x_i) - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x_i) \right\| \leq \Delta_{\mathbf{\Psi}(\mathbf{G})}, \tag{5}$$

where $\Delta_{\mathbf{\Psi}(\mathbf{G})} = \Delta_{\mathbf{\Psi}}(N)$ and δ_N are sequences satisfying $\Delta_{\mathbf{\Psi}}(N) \to 0$ and $\delta_N \to 0$ as $N \to \infty$.

Definition 1 formalizes the notion of transferability as a function whose output converges to the limit object operating on the manifold. As a consequence, a transferable function $\mathbf{\Psi}_{\mathbf{G}}$ can be trained or defined on finite graphs of one size and deployed on other graph instances—possibly of different sizes or resolutions—as long as the associated operators converge to the same manifold limit $\mathcal{L}$.

4.3 Transferable Graph Transformers

Under the transferability framework established above, we now turn to characterizing transferability for a Graph Transformer to its counterpart limit, a Manifold Transformer.

Definition 2 (Softmax Graph Transformer)**.** *A Graph Transformer (GT) layer is given by*

$$\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})_i = \frac{\sum_{j=1}^N \exp\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle \, \mathbf{V}\mathbf{x}_j}{\sum_{j=1}^N \exp\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle}, \tag{6}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices, respectively. The input to the GT is the output of a positional encoding module, $\mathbf{X} = \mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g) \in \mathbb{R}^{q \times N}$.

Definition 3 (Softmax Manifold Transformer)**.** *A Manifold Transformer (MT) layer is given by*

$$\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x) = \frac{\int_{\mathcal{M}} \exp\langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle \, \mathbf{V}f(y)\, d\mu(y)}{\int_{\mathcal{M}} \exp\langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle \, d\mu(y)}, \tag{7}$$

for all $x \in \mathcal{M}$, with the input $f \in L^2(\mu; \mathbb{R}^q)$ defined as the positional encoding of a manifold signal $g \in L^2(\mu; \mathbb{R}^p)$, that is, $f(x) = \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x)$, where $\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g) : \mathcal{M} \to \mathbb{R}^q$ is the manifold positional encoding.

The MT is a map $\mathbf{\Phi}_{\mathcal{M}} : L^2(\mu; \mathbb{R}^p) \to L^2(\mu; \mathbb{R}^q)$ resulting from the composition of f (the positional encoding) with manifold attention. The vector $f(x)$ corresponds to the output of the positional encodings in the manifold domain. The manifold attention of Equation (7) is the continuous analogue of Equation (6), with infinite attention coefficients for each point in the manifold. In a similar fashion, the manifold positional encoding $\mathbf{\Psi}_{\mathcal{M}}$ is the limit object of the graph positional encoding $\mathbf{\Psi}_{\mathbf{G}}$.

The following set of assumptions are required to ensure the convergence of GTs to MTs.

Assumption 1 (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 $\mathbf{x} \in \mathbb{R}^q$. We write $C_{\mathcal{T}} = C_Q C_K C_V$.*

Assumption 2 (Bounded and normalized Lipschitz positional encoding)**.** *For all $x, y \in \mathcal{M}$, the positional encoding function satisfies $\|\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x) - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(y)\|_2 \leq d_{\mathcal{M}}(x, y)$, where $d_{\mathcal{M}}$ is the geodesic distance on $\mathcal{M}$, and is bounded as $\|\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x)\|_2 \leq 1$. The graph positional encoding is also normalized with $\|\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)(x_i)\|_2 \leq 1$ for all $x_i \in X_N$.*

Assumption 2 is a mild assumption on the PE limit object. Assumption 1 can be seen as having regularized transformer operators. These can be realized in practice via penalty terms during the training process.

We can now present our main theorem on the convergence of GT to MT, which we will use to show the transferability of GT.

Theorem 1. *(Point-wise Convergence of GT to MT) Assume $\mathbf{\Psi}_{\mathbf{G}}$ is a transferable positional encoding with limit object $\mathbf{\Psi}_{\mathcal{M}}$, i.e., $\|\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)(x_i) - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x_i)\| \leq \Delta_{\mathbf{\Psi}(\mathbf{G})}$ for each $x_i \in X_N$, where $\mathcal{H}$ collects learnable parameters. Under Assumptions 1 and 2, for each node $x_i \in X_N$, the output difference between GT and MT, with probability at least $1 - \delta$, is bounded by*

$$\begin{aligned}\|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\|_2 &\leq \Delta_{\mathbf{\Phi}(\mathbf{G})} \\ &= \left[C_V + 4e^{2B}C_{\mathcal{T}}\right]\Delta_{\mathbf{\Psi}(\mathbf{G})} + \left[e^B C_V + 2e^{2B}C_{\mathcal{T}}\right] r, \end{aligned} \tag{8}$$

where r is the radius of the balls containing the partitions of $\mathcal{M}$ induced by X_N (Appendix B.1), which satisfies $r \leq A\,(\log N/N)^{1/d}$ if $d \geq 3$ and $r \leq A(\log N)^{3/4}/N^{1/2}$ if $d = 2$, with A a constant related to the geometry of $\mathcal{M}$, d the intrinsic dimension of the manifold, and $B := C_Q C_K$.

The proof of Theorem 1 is available in Appendix C.3. 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 the discretization term decaying at a rate $\mathcal{O}((\log N/N)^{1/d})$ and the remaining term decaying with the PE error $\Delta_{\mathbf{\Psi}(\mathbf{G})}$. The dependence of this result on the linear operator bounds $C_{\mathcal{T}}$ and C_V, as well as the bound on the positional encoding $\Delta_{\mathbf{\Psi}(\mathbf{G})}$, suggests that smoother linear operators on the transformer and positional encoding can improve the convergence rate.

Theorem 1 indicates that if the PEs are transferable, 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 MT's output is bounded. This implication is meaningful in practice 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 explicitly in the following corollary:

Corollary 2. *(Transferability of GTs) Let $\mathbf{G}_1$ and $\mathbf{G}_2$ be graphs constructed by points sampled from $\mathcal{M}$, with N_1, N_2 nodes respectively, and graph signals $\mathbf{X}_1$ and $\mathbf{X}_2$, and let $\|\cdot\|_{L^{1,2}(\mathcal{M})}$ be the norm defined in Appendix B.1. Then, it holds, with probability $1 - \delta$,*

$$\begin{aligned}\frac{1}{\mu(\mathcal{M})}||\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T}, \mathbf{X}_1) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T}, \mathbf{X}_2)||_{L^{1,2}(\mathcal{M})} &\leq \Delta_{\mathbf{\Phi}(\mathbf{G}_1)} + \Delta_{\mathbf{\Phi}(\mathbf{G}_2)} \\ &\quad + 4e^{2B}C_{\mathcal{T}}\,(r_1 + r_2)\,, \end{aligned} \tag{9}$$

where r_1 and r_2 are the partition radii of $\mathbf{G}_1$ and $\mathbf{G}_2$ (Appendix B.1), bounded as in Theorem 1.

Remark 2. *Theorem 1 and Corollary 2 are proven for a generalized version of the softmax GT presented in this section. The same proof strategy can be extended to other attention mechanism beyond softmax when the discrete operators could approximate a well-defined continuum operator. Furthermore, the framework extends to point clouds and meshes whose associated discrete operators and sampling schemes satisfy the assumed manifold convergence conditions.*

5 Convergence of Positional Encodings

Corollary 2 expresses a notion of transferability *inheritance* — if the positional encodings $\mathbf{\Psi_G}$ are transferable, then the following GT $\mathbf{\Phi_G}$ transfers as well. The choice of $\mathbf{\Psi_G}$ is the primary factor we can control in improving GT transferability via the PE convergence term $\Delta_{\mathbf{\Psi}(\mathbf{G})}$. In this section we compare the convergence bounds for two choices of $\mathbf{\Psi_G}$.

5.1 Eigenvector Positional Encoding Convergence

A common GT positional encoding consists of truncations of the graph Laplacian eigenspace (Dwivedi & Bresson, 2020). The analogue in the manifold domain is to truncate to a fixed set of eigenfunctions of the Laplace–Beltrami operator. Let $\{(\lambda_j, \boldsymbol{\phi}_j)\}_{j\geq 0}$ and $\{(\lambda_{j,N}, \boldsymbol{\phi}_{j,N})\}_{j=0}^{N-1}$ be the eigenpairs of $\mathcal{L}$ and $\mathbf{L}_N$, i.e., $\mathcal{L}\boldsymbol{\phi}_j = \lambda_j \boldsymbol{\phi}_j$ and $\mathbf{L}_N \boldsymbol{\phi}_{j,N} = \lambda_{j,N}\boldsymbol{\phi}_{j,N}$, with eigenvalues sorted in increasing order. The graph and manifold PEs can be defined as

Definition 4 (Eigenvector and Eigenfunction positional encodings)**.**

$$\mathbf{\Psi}_{\mathbf{G}}^{(K)}(\mathbf{L}_N)(x_i) := \begin{bmatrix}[\boldsymbol{\phi}_{1,N}]_i & \cdots & [\boldsymbol{\phi}_{K,N}]_i\end{bmatrix}^\top, \tag{10}$$

$$\mathbf{\Psi}_{\mathcal{M}}^{(K)}(\mathcal{L})(x) := \begin{bmatrix}\boldsymbol{\phi}_1(x) & \cdots & \boldsymbol{\phi}_K(x)\end{bmatrix}^\top. \tag{11}$$

Furthermore, denote their eigengaps by $\Gamma_{K,N}$ and Γ_K, respectively.

The graph eigenvectors are an approximation of the Laplace–Beltrami eigenfunctions. According to the Davis–Kahan theorem, this approximation is inversely proportional to the eigengap. This leads to unstable transferability, as we can see in the following theorem.

Theorem 3 (Convergence of truncated Laplacian eigenvectors to manifold limit)**.** *Fix $K > 0$ and let $\varepsilon_N \geq (\log N/N)^{1/(2d+8)}$. Then, there exist signs $r_{j,N} \in \{-1, 1\}$, $1 \leq j \leq K$, such that, with $\mathbf{R}_N := \mathrm{diag}(r_{1,N}, \ldots, r_{K,N})$, for every $x_i \in X_N$, with probability at least $1 - N^{-2}$,*

$$\left\| \mathbf{R}_N \mathbf{\Psi}_{\mathbf{G}}^{(K)}(\mathbf{L}_N)(x_i) - \mathbf{\Psi}_{\mathcal{M}}^{(K)}(\mathcal{L})(x_i) \right\|_2 \leq \Delta^{\mathrm{Lap}}_{\mathbf{\Psi}(\mathbf{G}_N)}(K) = \frac{\sqrt{K}\,\Omega_K}{\Gamma_K}\,\varepsilon_N^2, \tag{12}$$

where $\Gamma_K := \min_{1\leq j\leq K}(\lambda_{j+1} - \lambda_j) > 0$ is the eigengap of $\mathcal{L}$, and the constant Ω_K depends on λ_K, d, and the diameter, volume, injectivity radius, curvature and second fundamental form of $\mathcal{M}$.

The proof is in Appendix D.1. The convergence rate is inversely proportional to the eigengap of the Laplace–Beltrami operator, Γ_K. Consequently, for eigenvalues that are close, convergence is more unstable. In principle, the instability could be addressed by picking a value of K corresponding to large Γ_K. However, Γ_K is not available in practice, and so K is chosen based on $\Gamma_{K,N}$, the eigengap of the graph Laplacian, which requires a sample of at least order $(\log N/N)^{1/(2d+8)}$ for an accurate estimate. See Appendix D.2 for more discussion on eigengap approximations.

Finally, the sign matrix $\mathbf{R}_N$ aligns the signs of eigenvectors, given that $\boldsymbol{\phi}_{j,N}$ and $-\boldsymbol{\phi}_{j,N}$ are both eigenvectors of $\mathbf{L}_N$. This sign ambiguity poses a practical difficulty for PE methods that explicitly compute eigendecompositions. It is typically addressed by techniques such as augmenting training data with random eigenvector flips (Dwivedi & Bresson, 2020) or smoothening eigenvalues with a permutation equivariant function (Huang et al., 2024b).

5.2 GNN Positional Encoding Convergence

The challenges of eigenvector positional encodings, namely, their unstable convergence and sign ambiguity, can be avoided by using a smooth positional encoding that implicitly operates on eigenvectors, such as GNNs. We consider GNNs, whose size-transferability is well-established, and whose limit object is a manifold neural network (MNN), which can be defined as

Definition 5 (Manifold neural network). *One layer of an MNN maps the manifold signal g to*

$$\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x) = \sigma\Big[h(\mathcal{L})g(x)\Big], \tag{13}$$

where $h(\mathcal{L})$ applies the filter determined by the coefficients $\mathcal{H} = \{h_0, \ldots, h_{K-1}\} \subset \mathbb{R}$ based on Laplace-Beltrami operator $\mathcal{L}$, and σ is a pointwise nonlinearity.

We now present the convergence of GNNs to MNNs as follows.

Theorem 4 (Point-wise Convergence of GNN to MNN). *Let $\mathbf{\Psi}_{\mathbf{G}}$ and $\mathbf{\Psi}_{\mathcal{M}}$ be a GNN and MNN with the same filters $\mathcal{H}$. Under the normalized Lipschitz signal and nonlinearity assumptions, and the non-amplifying and spectral continuity assumptions on the filters, for every node $x_i \in X_N$, with probability at least $1 - \delta$,*

$$\|\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)(x_i) - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x_i)\|_2 \leq \Delta^{\mathrm{GNN}}_{\mathbf{\Psi}(\mathbf{G}_N)} = C\varepsilon_N^2 + \frac{\pi^2}{6}\sqrt{\frac{\log(1/\delta)}{N}}, \tag{14}$$

where C depends on the geometry of the manifold and scales with the filter constant C_L.

The GNN bound of Theorem 4 does not depend on the truncated eigengap Γ_K. Instead we have the GNN Lipschitz constant C_L, which can be controlled via regularization. Therefore, GNNs are more stable than eigenvectors in the sense of size-transferability. In contrast to the eigenvector convergence bound of Theorem 3, the GNN bound does not depend on the spectral gap. This result is adapted to our setting from the GNN to MNN convergence results in Wang et al. (2025).

Finally, in Appendix E we present Corollaries 12 and 13 to state the explicit transferability on GTs with eigenvector and GNN PEs, followed by application of Theorems 3 and 4 into Corollary 2.

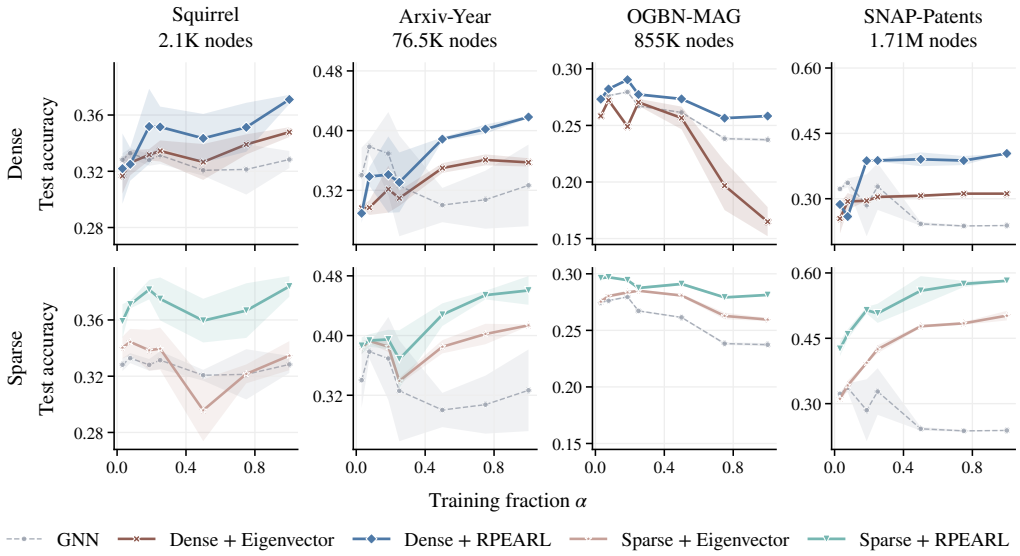


Figure 1: Transferability of GTs with RPEARL and Laplacian PEs measured by accuracy (higher is better ↑). The x-axis represents the train graph sizes as a proportion of the largest graph (α), and the y-axis is the test accuracy at the full-sized graph. The top row shows dense GTs, and the bottom row shows sparse GTs. The GNN baseline appears in both rows for reference. Shaded regions show the standard deviation over 3 runs. The titles show dataset name and largest graph size.

6 Experiments

6.1 Transferability on graph datasets

Corollary 2 predicts a shrinking gap between GTs and MTs as graph sizes grows. In Figure 1, from the full graph, we first subsample a large test graph $\mathbf{G}_{\mathrm{TST}}$ with N_{TST} nodes. We then build training

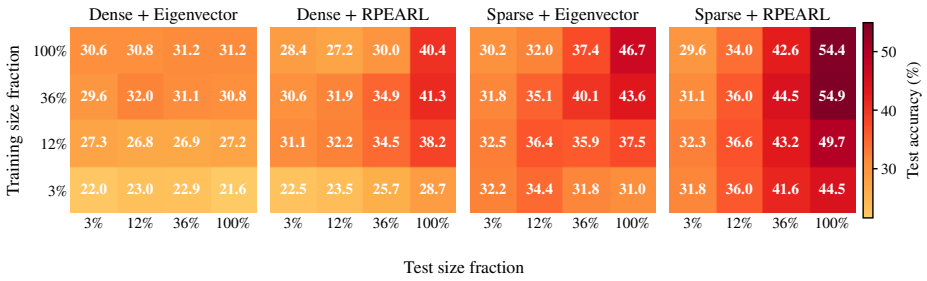


Figure 2: Transferability heatmaps on SNAP-Patents. The x-axis is the ratio of testing graph size to the full graph size while y-axis is the ratio of training graph sizes. The color represents the accuracy.

graphs by randomly sampling $N_{\text{TR}} = \alpha N_{\text{TST}}$ nodes from $\mathbf{G}_{\text{TST}}$ and taking the induced subgraphs, for training fractions $\alpha \in \{0.03, 0.08, 0.18, 0.25, 0.50, 0.75, 1.0\}$. We train each GT on its subgraph and evaluate on $\mathbf{G}_{\text{TST}}$, expecting the performance to improve as α increases and gradually approach that of a GT trained on the full test graph ($\alpha = 1$). In Figure 2, we subsample both training and testing graphs with different ratios, to further inspect the relationship between training sizes, testing sizes and accuracy.

Datasets. We evaluate node classification on Squirrel Rozemberczki et al. (2021), ArXiv-year (Lim et al., 2021), OGBN-MAG (Wang et al., 2020), and SNAP-Patents (Lim et al., 2021). Additional graph classification results on Reddit-Binary (Yanardag & Vishwanathan, 2015), appear in Appendix G.1.

Models. We compare dense GTs (Definition 2) and sparse GTs with k-hop attention masks (Appendix E.4), using RPEARL convolutional PEs (183) or Laplacian eigenvector PEs (10). Appendix G.4 includes additional PEs (Random Walk PEs and other GNNs), and backbones (Exphormer (Shirzad et al., 2023) and GraphGPS (Rampášek et al., 2022)).

Convolutional PEs yield more stable transferability than eigenvector PEs. Our empirical results are consistent with Theorems 3 and 4 and show that GTs inherit the more stable convergence of GNN-based PEs. Figure 1 shows that performance with RPEARL PEs improves as the training graph grows and then plateaus, while performance with Laplacian PEs can be unstable and degrade. On OGBN-MAG, the accuracy of the dense GT with Laplacian PEs peaks near $\alpha = 0.25$ and then drops sharply as the training graph grows. We do not observe this degradation with RPEARL PEs. Figure 2 shows accuracy across train and test graph sizes is robust. We observe that with the same training fraction, RPEARL PEs achieve better accuracy when transferred to larger testing graphs both in dense and sparse GT settings compared to Laplacian PE. Full observation of heatmap results can be found in Figure 6 in Appendix G.1. This indicates that GTs with RPEARL trained on smaller graphs can transfer effectively to larger graphs without retraining, supporting the use of smaller training graphs to reduce computational cost.

Sparse GTs preserve transferability at a fraction of the cost. Empirically, sparse GTs transfer at least as well as dense GTs while training up to $100\times$ faster, with speedups that grow with graph size (Appendix G.3). In Figure 1, sparse GT with RPEARL PEs exhibit superior generalization across models, with the largest margin on SNAP-Patents (0.58 vs. 0.51 for the next best model). On Squirrel and OGBN-MAG, sparse GTs with RPEARL PEs are close to their $\alpha = 1$ accuracy even at the smallest training fractions, whereas the dense GT needs substantially larger training graphs to reach comparable accuracy. Figure 2 further supports this advantage: at a fixed training graph size, sparse GTs sustain high accuracy as the testing graph grows, including when it exceeds the training graph size. We conjecture that the k-hop mask supplies local structure that partially compensates for the instability of eigenvector PEs.

6.2 TERRAIN GRAPH APPLICATION

A suitable application of our theoretical results emerges in the domain of distance estimation over terrain graphs. We follow the setup of (Chen et al., 2025), a metric learning problem over grid graphs to estimate shortest path distances. Each dataset comprises a high-resolution grid graph $\mathbf{G}$. We downsample $\mathbf{G}$ into coarser grid graphs $\mathbf{G}_r$ using strides $r \in \{4, 5, \ldots, 40\}$. For each $\mathbf{G}_r$, we train Sparse GT with an MSE loss on the L_1 approximation of node embeddings against its ground truth SPDs. After training, we evaluate Sparse GT on the full resolution $\mathbf{G}$ on a new set of sample points. For hyperparameters and more implementation details, see Appendix G.5.

We present the results on the Norway (Kartverket (Norwegian Mapping Authority), 2025) and Philadelphia (Chen et al., 2025) datasets in Figure 3 and provide a GNN baseline for reference. We observe that both architectures show strong transferability, achieving low errors on all training fractions. In particular, we observe that Sparse GT trained with as few as $10,000$ nodes shows performance comparable to training with an order of magnitude larger graphs. When averaged across fractions, we observe a 25.2% NMAE improvement relative to GNNs in Norway, and a 26.4% improvement in Philadelphia.

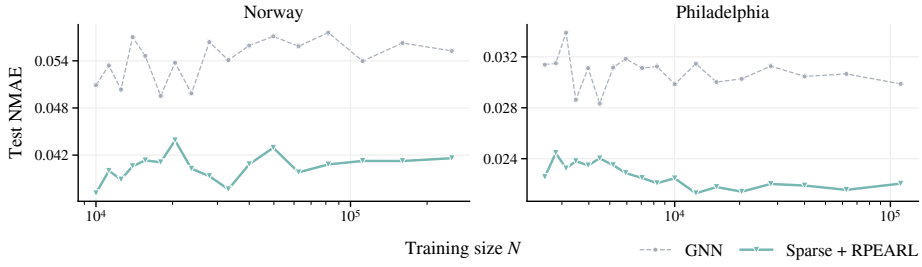


Figure 3: Transferability on terrain graphs measured by normalized mean absolute error (NMAE), lower is better ↓. The x-axis corresponds to the number of nodes for each training resolution, and the y-axis is the NMAE on the highest resolution test graph.

7 CONCLUSIONS

We have presented a framework showing that Graph Transformers endowed with transferable positional encodings inherit their transferability. We also provided empirical verification that various GTs exhibit competitive performance on large graphs when trained on smaller graphs. These results are valuable for applications dealing with large graphs, where data collection and training costs are a major challenge. This notion of transferability inheritance can be applied to other graph limit objects, for example, random graphs sampled from graphons. Furthermore, our results motivate future research directions into other structural, relative and absolute graph positional encodings that may be proven to be transferable if they satisfy the transferability hypothesis of Theorem 1.

8 LIMITATIONS

This work is scoped to the assumptions established in Section 4. We have provided commentary discussing why we do not believe these assumptions are restrictive in practice. Additionally, we assume that all the graph datasets used in this work are sampled from an underlying, low-dimensional manifold, commonly known as the manifold hypothesis. There is empirical evidence suggesting that the spectrum of graph datasets is indeed supported on a low-dimensional subspace, for instance see (Wang et al., 2025), Appendices H, I.

AI USE STATEMENT

In this work, we have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets, assist with translation are not applicable to this work. Additionally, we used generative AI tools for edit a research paper to improve readability, identify relevant literature, formulate questions for surveys or interviews, create or modify scientific figures or images, suggest experimental parameters, and create or edit software code. We have reviewed all AI-assisted work. All suggested revisions to improve the readability were reviewed and edited by the authors. LLM-assisted code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 38546–38556. Curran Associates, Inc., 2022.

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*, pp. 1–71, 2009.

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.

Mitchell Black, Zhengchao Wan, Gal Mishne, Amir Nayyeri, and Yusu Wang. Comparing graph transformers via positional encodings. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=va3r3hSA6n.

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.

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.

Edoardo Calvello, Nikola B. Kovachki, Matthew E. Levine, and Andrew M. Stuart. Continuum attention for neural operators. *Journal of Machine Learning Research*, 26(300):1–52, 2025. URL https://jmlr.org/papers/v26/24-0879.html.

Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5533–5566. PMLR, 2024. URL https://proceedings.mlr.press/v235/canturk24a.html.

Juan Cerviño, Luana Ruiz, and Alejandro Ribeiro. Learning by Transference: Training Graph Neural Networks on Growing Graphs. *IEEE Transactions on Signal Processing*, 71:233–247, 2023. ISSN 1053-587X, 1941-0476. doi: 10.1109/TSP.2023.3242374.

Samantha Chen, Pankaj K. Agarwal, and Yusu Wang. De-coupled NeuroGF for Shortest Path Distance Approximations on Large Terrain Graphs. In *Forty-Second International Conference on Machine Learning*, June 2025.

Krzysztof Choromanski, Han Lin, Haoxian Chen, Tianyi Zhang, Arijit Sehanobish, Valerii Likhosherstov, Jack Parker-Holder, Tamas Sarlos, Adrian Weller, and Thomas Weingarten. From block-Toeplitz matrices to differential equations on graphs: Towards a general theory for scalable

masked Transformers. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 3962–3983. PMLR, 2022. URL https://proceedings.mlr.press/v162/choromanski22a.html.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4ff87ac8f2f86bd471e25bed6eb69d77-Abstract-Conference.html.

Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017.

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.

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. 2021. URL https://openreview.net/forum?id=wTTjnvGphYj.

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.

James Flora, Mitchell Black, Weng-Keen Wong, and Amir Nayyeri. Understanding truncated positional encodings for Graph Neural Networks. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.13671.

Nicolás García Trillos, Moritz Gerlach, Matthias Hein, and Dejan Slepčev. Error estimates for spectral convergence of the graph laplacian on random geometric graphs toward the laplace–beltrami operator, 2020.

Simon Geisler, Yujia Li, Daniel J. Mankowitz, Ali Taylan Cemgil, Stephan Günnemann, and Cosmin Paduraru. Transformers Meet Directed Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 11144–11172. PMLR, 2023. URL https://proceedings.mlr.press/v202/geisler23a.html.

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

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

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Xinting Huang, Andy Yang, Satwik Bhattamishra, Yash Sarrof, Andreas Krebs, Hattie Zhou, Preetum Nakkiran, and Michael Hahn. A Formal Framework for Understanding Length Generalization in Transformers. In *The Thirteenth International Conference on Learning Representations*, October 2024a.

Yinan Huang, William Lu, Joshua Robinson, Yu Yang, Muhan Zhang, Stefanie Jegelka, and Pan Li. On the stability of expressive positional encodings for graphs. In *The Twelfth International Conference on Learning Representations*, 2024b.

Charilaos Kanatsoulis, Evelyn Choi, Stefanie Jegelka, Jure Leskovec, and Alejandro Ribeiro. Learning Efficient Positional Encodings with Graph Neural Networks. In *The Thirteenth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AWg2tkbydO.

Kartverket (Norwegian Mapping Authority). National elevation model (dem) of norway, 2025. URL https://www.geonorge.no/. Accessed: 2026-01-28.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan, Payel Das, and Siva Reddy. The Impact of Positional Encoding on Length Generalization in Transformers. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Nicolas Keriven and Samuel Vaiter. What functions can Graph Neural Networks compute on random graphs? the role of Positional Encoding. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper/2023/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html.

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.

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.

Ron Levie, Wei Huang, Lorenzo Bucci, Michael Bronstein, and Gitta Kutyniok. Transferability of spectral graph convolutional neural networks. *Journal of Machine Learning Research*, 22(272): 1–59, 2021.

Eitan Levin, Yuxin Ma, Mateo Díaz, and Soledad Villar. On transferring transferability: Towards a theory for size generalization. In *Advances in Neural Information Processing Systems*, 2025.

Hongkang Li, Meng Wang, Tengfei Ma, Sijia Liu, Zaixi Zhang, and Pin-Yu Chen. What improves the generalization of Graph Transformers? a theoretical dive into the self-attention and positional encoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 28784–28829. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bo.html.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, pp. 19565–19594. PMLR, 2023.

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 *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, pp. 20887–20902. arXiv, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ae816a80e4c1c56caa2eb4e1819cbb2f-Abstract.html.

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. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Q-UHqMorzil.

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)*, pp. 1057–1066. IEEE, 2024. ISBN 979-8-3503-6248-0. doi: 10.1109/BigData62323.2024.10825327. URL https://ieeexplore.ieee.org/document/10825327/.

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 *Proceedings of the 40th International Conference on Machine Learning*, pp. 23321–23337. PMLR, 2023. URL https://proceedings.mlr.press/v202/ma23c.html.

Sohir Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: an extended graphon approach. *Applied and Computational Harmonic Analysis*, 63:48–83, 2023.

Sohir Maskey, Gitta Kutyniok, and Ron Levie. Generalization bounds for message passing networks on mixture of graphons. *SIAM Journal on Mathematics of Data Science*, 7(2):802–825, 2025.

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

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

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient Context Window Extension of Large Language Models. In *The Twelfth International Conference on Learning Representations*, October 2023.

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.

Isaac Reid, Arijit Sehanobish, Cederik Höfs, Bruno Mlodozeniec, Leonhard Vulpius, Federico Barbero, Adrian Weller, Krzysztof Choromanski, Richard E. Turner, and Petar Veličković. Rotary position encodings for graphs. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.22259.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

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.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Graph neural networks: Architectures, stability, and transferability. *Proceedings of the IEEE*, 109(5):660–682, 2021.

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

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.

Timo Stoll, Luis Müller, and Christopher Morris. Generalizable insights for Graph Transformers in theory and practice. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/f79df6cbc6e5f708440004fad7ef64cc-Abstract-Conference.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010. Curran Associates Inc., 2017. ISBN 978-1-5108-6096-4.

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

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1): 396–413, 2020.

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

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Stability to deformations of manifold filters and manifold neural networks. *IEEE Transactions on Signal Processing*, pp. 1–15, 2024b. doi: 10.1109/TSP.2024.3378379.

Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro. A manifold perspective on the statistical generalization of graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62467–62504. PMLR, 13–19 Jul 2025.

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, December 2023.

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. 2022. doi: 10.48550/arXiv.2201.08821. URL http://arxiv.org/abs/2201.08821.

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, pp. 1365–1374. Association for Computing Machinery, 2015. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2783417. URL https://dl.acm.org/doi/10.1145/2783258.2783417.

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.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10737–10745, 2021.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers Can Achieve Length Generalization But Not Robustly. Number arXiv:2402.09371. arXiv, February 2024. doi: 10.48550/arXiv.2402.09371.

A Related Work

Transferability and Generalization via Limit Models. The convergence of GNNs to manifold neural networks (MNNs) has been leveraged extensively as an analytical tool to establish the transferability, stability, and statistical generalization of graph neural networks (Wang et al., 2024a;b; 2025). Transferability has also been explored with graphons as a limit model for graphs (Ruiz et al., 2020; Maskey et al., 2025; 2023). Keriven & Vaiter (2023) studied the limit function space of GNNs over random graphs, as well as the role of positional encodings. More recently, Levin et al. (2025) studied a general framework for transferability by relating small graph instances to limit objects. Beyond graph models, Calvello et al. (2025) study the attention on general function spaces and connect discrete attention to continum operators.

Efficient training on large graphs. GraphSAGE (Hamilton et al., 2017) scales GNNs to large-scale graphs by training on minibatches of neighborhood subgraphs. Lin et al. (2024) leverages GraphSAGE-style sampling. Based on graphon convergence results, Cerviño et al. (2023) proposes training on growing graphs. Deng et al. (2024) improves scalability through linear local-to-global attention mechanism.

Length Generalization in Transformers. Length generalization in large language models has been studied both empirically (Zhou et al., 2024; Anil et al., 2022) and theoretically (Huang et al., 2024a). It has been argued that PEs play an important role in achieving robust length generalization (Kazemnejad et al., 2023; Peng et al., 2023). Yet, the characterization and necessary conditions for length generalization remain an open area of research. Li et al. (2023) presented a generalization analysis for general Transformers without considering graph structural information. Statistical generalization of graph transformers have also been studied in recent years with relative positional encodings (Li et al., 2024).

GNN-Transformer Hybrids. Several works combine message-passing GNNs with global attention into hybrid "local-global" models, including GraphGPS (Rampášek et al., 2022), SGFormer (Wu et al., 2023), GraphTrans (Wu et al., 2022), and UnifiedGT (Lin et al., 2024). Our theoretical results on the transferability of GTs also cover these hybrid architectures.

GT Positional Encodings. Beyond Laplacian eigenvectors, other additive encodings include learned spectral encodings drawn from the Laplacian spectrum (SAN) (Kreuzer et al., 2021) or learned mappings of the random walk matrix (Dwivedi et al., 2021). Spectral encoders have also been used to address the eigenvector signs and eigenspace basis changes (Lim et al., 2023) as well as the stability under perturbations (Huang et al., 2024b). Learnable encoders include pretrained foundation graph encoder with positional and structural representations (Cantürk et al., 2024) as well as GNN-based positional encodings (Kanatsoulis et al., 2024). Relative PEs encode pairwise information, such as GraphiT (diffusion kernels) (Mialon et al., 2021), the Graphormer family (shortest-path and centrality biases) (Ying et al., 2021), random walk approaches (Ma et al., 2023; Geisler et al., 2023), or GKAT (heat-kernel-based) (Choromanski et al., 2022). Mask-based relative encodings replace quadratic attention with structured sparsity to trade off globality and scalability, for example, Exphormer (local neighborhood + expander graphs) (Shirzad et al., 2023), and UnifiedGT (k-hop neighborhood) (Lin et al., 2024). We refer to (Black et al., 2024) for an in-depth analysis of the properties of absolute and relative positional encodings on GTs. Recent works further study the expressive limits with truncations (Flora et al., 2026) and combinations with rotary positional encodings (Reid et al., 2026). Stoll et al. (2025) introduce the Generalized-Distance Transformer and analyze its representation power in relation to attention and positional encodings.

B Preliminaries

B.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 $\mathbf{X} \in \mathbb{R}^{q \times N}$ can be viewed as a discretization of the continuous manifold signal f evaluated at points

X_N, that is

$$\mathbf{S}_N f = \mathbf{X}, \tag{15}$$

where $\mathbf{S}_N : L^2(\mu) \to L_2(X_N)$ is called the sampling operator. The sample X_N induces a decomposition of the manifold (García Trillos et al., 2020), $\{V_i\}_{i=1}^N$ 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 geodesic distance $d_{\mathcal{M}}$ on $\mathcal{M}$.

Let $\mu_N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ be 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,

$$d_\infty(\mu, \mu_N) = \min_{T:T\#\mu=\mu_N} \operatorname*{ess\,sup}_{x\in\mathcal{M}} d_{\mathcal{M}}(x, T(x)). \tag{16}$$

Here, $T\#\mu$ denotes the pushforward measure, i.e., $\mu(T^{-1}(V)) = \mu_N(V)$ holds for every subset $V \subseteq X_N$, and $V_i = T^{-1}(\{x_i\})$.

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

$$(\mathbf{I}_N\mathbf{X})(x) = \sum_{i=1}^N [\mathbf{X}]_i \mathbb{1}_{x\in V_i}, \tag{17}$$

where $\mathbf{I}_N : L_2(X_N) \to L^2(\mu)$ denotes the interpolation operator. To compare induced signals of graphs with different numbers of nodes, we use the norm $\|h\|_{L^{1,2}(\mathcal{M})} := \int_{\mathcal{M}} \|h(x)\|_2 \, d\mu(x)$ for $h \in L^2(\mu; \mathbb{R}^q)$, i.e., the integral of the pointwise Euclidean norm.

B.2 Spectral convergence of graph Laplacians

We recall the pointwise spectral convergence of the graph Laplacian to the Laplace–Beltrami operator from (Dunson et al., 2021).

Proposition 5. *(Dunson et al., 2021)[Theorem 4] For a sufficiently small $\varepsilon_N > 0$, if N is sufficiently large so that $\varepsilon_N \geq \left(\frac{\log N}{N}\right)^{\frac{1}{2d+12}}$, then with probability greater than $1 - N^{-2}$, for all $0 \leq i < J$,*

$$|\lambda_{i,N} - \lambda_i| \leq \Omega_1 \varepsilon_N^2, \tag{18}$$

with Ω_1 is 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.

C Proofs of Graph Transformer Transferability

C.1 Generalized Graph and Manifold Transformers

In Section 4 we discuss softmax-based GTs and MTs. These come from the following generalized definitions.

Definition 6 (Generalized GT)**.** *A GT layer is given by*

$$\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})_i = \frac{\sum_{j=1}^N \gamma_{ij} \mathbf{V}\mathbf{x}_j}{\sum_{j=1}^N \gamma_{ij}}, \tag{19}$$

where $\gamma_{ij} = \gamma(\mathbf{x}_i, \mathbf{x}_j)$, and $\gamma : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}_+$ is called the kernel function.

Definition 7 (Manifold Transformer)**.** *A Manifold Transformer (MT) layer is given by*

$$\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x) = \frac{\int_{\mathcal{M}} \tilde{\gamma}_{xy} \mathbf{V} f(y)\, d\mu(y)}{\int_{\mathcal{M}} \tilde{\gamma}_{xy}\, d\mu(y)}, \tag{20}$$

for all $x \in \mathcal{M}$, with $\tilde{\gamma}_{xy} := \gamma(f(x), f(y))$, and the input $f \in L^2(\mu; \mathbb{R}^q)$ defined as the positional encoding of a manifold signal $g \in L^2(\mu; \mathbb{R}^p)$, that is, $f(x) = \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x)$.

The Manifold Transformer is a map $\mathbf{\Phi}_{\mathcal{M}} : L^2(\mu; \mathbb{R}^p) \to L^2(\mu; \mathbb{R}^q)$ resulting from the composition of f (the positional encoding) with manifold attention. The vector $f(x)$ corresponds to the output of the positional encodings in the manifold domain. The manifold attention of Equation (20) is the continuous analogue of Equation (19), with infinitely many attention coefficients for each point in the manifold.

Remark 3. *The softmax graph attention of Equation* (1) *is the particular case of Definition 6 where attention coefficients are given by $\gamma_{ij} = \exp \langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle$.*

Our convergence results hold for any kernel satisfying the following assumption.

Assumption 3 (Valid kernel function)**.** *The kernel function γ_{xy} satisfies the following: (i) Non-negative: For any $x, y \in \mathcal{M}$, $\gamma_{xy} \geq 0$. (ii) Bounded: $\gamma_{xy} \leq C$ for some constant C, (iii) For $x, y, x', y' \in \mathcal{M}$ and signal f, it holds that $\|\gamma_{xy} - \gamma_{x'y'}\| \leq L_\gamma \big(\|f(x) - f(x')\| + \|f(y) - f(y')\| \big)$.*

The exponential kernel of softmax attention, $\gamma_{xy} = \exp \langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle$, satisfies Assumption 3: it is nonnegative, bounded, and Lipschitz with constant $L_\gamma = C_Q C_K e^B$, where $B := C_Q C_K$, as we prove in Proposition 6.

A comment on manifold attention. Notice that, for each manifold point x, the normalized manifold attention weights $\gamma_x(y) := \tilde{\gamma}_{xy} / \int_{\mathcal{M}} \tilde{\gamma}_{xy'}\, d\mu(y')$ form a probability density over $\mathcal{M}$. Thus, manifold attention is an expectation over $\mathbf{V}f(y)$ with distribution γ_x supported on the manifold, $\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x) = \int_{\mathcal{M}} \mathbf{V}f(y) \gamma_x(y)\, d\mu(y) = \mathbb{E}_{y \sim \gamma_x} [\mathbf{V}f(y)]$. Therefore, the special case of softmax attention can be interpreted as the expectation over values with distribution γ_x a Gibbs measure with energies $E_x(y) = -\langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle$.

C.2 Auxiliary lemmas and propositions

In this section we state a series of lemmas and propositions that will be useful for the proof of Theorem 1. Throughout the proofs, let $B := C_Q C_K$, so that, by Assumptions 1 and 2 (bounded operators and normalized encodings), $|\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle| \leq B$, and $\sup_{u,v \in \mathcal{M}} |\langle \mathbf{Q}f(u), \mathbf{K}f(v) \rangle| \leq B$.

Proposition 6 (Exponential Kernel)**.** *Assumption 3 holds for the kernel $\gamma(x, y) = \gamma_{xy} = \exp [\langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle]$.*

Proof. The exponential function is nonnegative, so (i) holds. For (ii), since the signals are bounded, we can bound the kernel by $\exp [\langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle] \leq e^B$, Finally, for (iii), let $x, y, x', y' \in \mathcal{M}$, then we have

$$\begin{aligned}
&\| \exp [\langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle] - \exp [\langle \mathbf{Q}f(x'), \mathbf{K}f(y') \rangle] \| \\
&\quad \leq e^B \Big| [\langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle] - [\langle \mathbf{Q}f(x'), \mathbf{K}f(y') \rangle] \Big| && (21) \\
&\quad = e^B \Big| [\langle \mathbf{Q}(f(x) - f(x')), \mathbf{K}f(y) \rangle] - [\langle \mathbf{Q}f(x'), \mathbf{K}(f(y') - f(y)) \rangle] \Big| && (22) \\
&\quad \leq e^B C_Q C_K \big[\|f(x) - f(x')\| + \|f(y) - f(y')\| \big]. && (23)
\end{aligned}$$

where in (21) we use the mean value theorem, in (22) we add and subtract $\langle \mathbf{Q}f(x'), \mathbf{K}f(y) \rangle$, and in (23) we bound the linear operators, use $\|f\| \leq 1$ (Assumption 2), and apply the triangle inequality. □

Lemma 7. *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*

$$|\langle \mathbf{Q}f(x), \mathbf{K}f(x_j) \rangle - \langle \mathbf{Q}f(x), \mathbf{K}f(y) \rangle| \leq C_Q C_K\, r \tag{24}$$

Proof.

$$|\langle \mathbf{Q}f(x), \mathbf{K}f(x_j)\rangle - \langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle| = |\langle \mathbf{Q}f(x), \mathbf{K}\left(f(x_j) - f(y)\right)\rangle| \tag{25}$$
$$\leq C_Q\|f(x)\| \cdot C_K\|f(x_j) - f(y)\| \tag{26}$$
$$\leq C_Q C_K\|f(x_j) - f(y)\| \tag{27}$$
$$\leq C_Q C_K d_{\mathcal{M}}(x_j, y) \tag{28}$$
$$\leq C_Q C_K\, r \tag{29}$$

Where in (26) we apply the bound on the linear operators $\mathbf{Q}$ and $\mathbf{K}$, in (27) we apply the bound $\|f(x)\| \leq 1$ from Assumption 2, in (28) we apply Assumption 2, and in (29) we use the fact that $y \in V_j$, therefore $d_{\mathcal{M}}(y, x_j) \leq r$. □

For notational brevity, onwards 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 $\tilde{\gamma}_{ij} = \exp[\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle]$.

Lemma 8. *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*

$$|\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| \leq e^B C_Q C_K\, r \tag{30}$$

Proof.

$$|\tilde{\gamma}_{ij} - \tilde{\gamma}_{iy}| = |\exp\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle - \exp\langle \mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle| \tag{31}$$
$$\leq e^B\, |\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle - \langle \mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle| \tag{32}$$
$$\leq e^B C_Q C_K\, r \tag{33}$$

where $B = C_Q C_K$ as in Appendix C.2. 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 7. □

Lemma 9. *For each $x_i, x_j \in X_N$ it holds that*

$$|\gamma_{ij} - \tilde{\gamma}_{ij}| \leq 2e^B C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})}. \tag{34}$$

Proof.

$$|\gamma_{ij} - \tilde{\gamma}_{ij}| = \Big|\exp\big(\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle\big) - \exp\big(\langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle\big)\Big| \tag{35}$$
$$\leq e^B \Big|\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j\rangle - \langle \mathbf{Q}f(x_i), \mathbf{K}f(x_j)\rangle\Big| \tag{36}$$
$$\leq e^B \Big(\big|\langle \mathbf{Q}(\mathbf{x}_i - f(x_i)), \mathbf{K}\mathbf{x}_j\rangle\big| + \big|\langle \mathbf{Q}f(x_i), \mathbf{K}(\mathbf{x}_j - f(x_j))\rangle\big|\Big) \tag{37}$$
$$\leq e^B \Big(\|\mathbf{Q}\|\,\|\mathbf{K}\|\,\|\mathbf{x}_i - f(x_i)\|\,\|\mathbf{x}_j\| + \|\mathbf{Q}\|\,\|\mathbf{K}\|\,\|f(x_i)\|\,\|\mathbf{x}_j - f(x_j)\|\Big) \tag{38}$$
$$\leq 2e^B C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})}. \tag{39}$$

In (36) we apply the mean value theorem $|e^a - e^b| \leq e^{\max\{a,b\}}|a - b|$ and upper-bound $\max\{a, b\} \leq B$. In (37) we add and subtract $\langle \mathbf{Q}f(x_i), \mathbf{K}\mathbf{x}_j\rangle$ and use bilinearity and the triangle inequality. In (38) we apply Cauchy–Schwarz and boundedness of the linear operators. In (39) we use Assumption 2 and the transferable-PE hypothesis of Theorem 1. □

Lemma 10. *Let $x, y \in \mathcal{M}$, with $d_{\mathcal{M}}(x, x_i) \leq r$. Then, it holds that*

$$|\tilde{\gamma}_{iy} - \tilde{\gamma}_{xy}| \leq e^B C_{QK} r \tag{40}$$

Proof.

$$|\tilde{\gamma}_{iy} - \tilde{\gamma}_{xy}| = |\exp\langle \mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle - \exp\langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle| \tag{41}$$
$$\leq e^B\, |\langle \mathbf{Q}f(x_i), \mathbf{K}f(y)\rangle - \langle \mathbf{Q}f(x), \mathbf{K}f(y)\rangle| \tag{42}$$
$$= e^B |\langle \mathbf{Q}\left[f(x_i) - f(x)\right], \mathbf{K}f(y)\rangle| \tag{43}$$
$$\leq e^B C_{QK} r. \tag{44}$$

where in (42) we use $|e^a - e^b| \leq e^{\max\{a,b\}}|a-b|$ and upper-bound $\max\{a,b\} \leq B$, in (43) we rearrange the terms, and in (44) we use the bounds on linear operators, bounded manifold signal, normalized Lipschitz of f, and $d_{\mathcal{M}}(x, x_i) \leq r$. □

C.3 Proof of Theorem 1

Theorem 1 bounds the convergence of a Graph Transformer with transferable PEs to a Manifold Transformer with the corresponding limit PEs. In the proof we combine linear operator constants $C_{QK} := C_Q C_K$ and $C_Q C_K C_V = C_{\mathcal{T}}$.

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

$$\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) = \frac{\sum_{j=1}^{N} \exp[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle] \mathbf{V}\mathbf{x}_j}{\sum_{j=1}^{N} \exp[\langle \mathbf{Q}\mathbf{x}_i, \mathbf{K}\mathbf{x}_j \rangle]} \tag{45}$$

We use the attention coefficients γ_{ij}, $\tilde{\gamma}_{iy}$ and $\tilde{\gamma}_{ij}$ defined in Appendix C.2. Furthermore, when necessary we will abbreviate the denominator terms as $D_{\mathbf{G}} = \sum_{j=1}^{N} \gamma_{ij}$, $D_{\mathcal{M}} = \int_{\mathcal{M}} \tilde{\gamma}_{iy} d\mu(y)$, and $\bar{D}_{\mathcal{M}} = \sum_{j=1}^{N} \tilde{\gamma}_{ij}$.

Thus we have

$$\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) = \frac{1}{D_{\mathbf{G}}} \sum_{j=1}^{N} \gamma_{ij} \mathbf{V}\mathbf{x}_j \tag{46}$$

$$\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i) = \frac{1}{D_{\mathcal{M}}} \int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \tag{47}$$

We will introduce an auxiliary term that evaluates the MT kernel and values at the sampled points, denoted by

$$\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) = \frac{\sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y)}{\sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} d\mu(y)}. \tag{48}$$

Notice that since the integrand does not depend on y, we can simplify $\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i)$ to:

$$\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) = \frac{\sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y)}{\sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} d\mu(y)} \tag{49}$$

$$= \frac{\sum_{j=1}^{N} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) \cdot \mu(V_j)}{\sum_{j=1}^{N} \tilde{\gamma}_{ij} \cdot \mu(V_j)} \tag{50}$$

$$= \frac{\frac{1}{N} \sum_{j=1}^{N} \tilde{\gamma}_{ij} \mathbf{V} f(x_j)}{\frac{1}{N} \sum_{j=1}^{N} \tilde{\gamma}_{ij}} \tag{51}$$

$$= \frac{1}{\bar{D}_{\mathcal{M}}} \sum_{j=1}^{N} \tilde{\gamma}_{ij} \mathbf{V} f(x_j), \tag{52}$$

The output difference for node i can be decomposed as

$$\begin{aligned} &\|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \\ &= \|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) + \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \qquad (53) \\ &\leq \|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i)\| + \|\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \qquad (54) \end{aligned}$$

To Equation (53) we add and subtract the induced manifold signal term, and in (53) to (54) we use the triangle inequality.

We will now bound the first term of (54) as

$$\left|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T},\mathbf{X})(x_i)-\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T},\mathbf{X})(x_i)\right|=\left|\frac{1}{D_{\mathbf{G}}}\sum_{j=1}^{N}\gamma_{ij}\mathbf{V}\mathbf{x}_j-\frac{1}{\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}\tilde{\gamma}_{ij}\mathbf{V}f(x_j)\right|. \tag{55}$$

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

$$\begin{aligned}
&\left\|\frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}\mathbf{x}_j-D_{\mathbf{G}}\tilde{\gamma}_{ij}\mathbf{V}f(x_j)\right\| \\
&=\left\|\frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}\mathbf{x}_j-\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}f(x_j)+\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}f(x_j)-D_{\mathbf{G}}\tilde{\gamma}_{ij}\mathbf{V}f(x_j)\right\| &(56)\\
&\leq\left\|\frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}\mathbf{x}_j-\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}f(x_j)\right\|+ \\
&\qquad\qquad\left\|\frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}\bar{D}_{\mathcal{M}}\gamma_{ij}\mathbf{V}f(x_j)-D_{\mathbf{G}}\tilde{\gamma}_{ij}\mathbf{V}f(x_j)\right\| &(57)\\
&=\left\|\frac{1}{D_{\mathbf{G}}}\sum_{j=1}^{N}\gamma_{ij}\mathbf{V}(\mathbf{x}_j-f(x_j))\right\|+\left\|\frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}}\sum_{j=1}^{N}(\bar{D}_{\mathcal{M}}\gamma_{ij}-D_{\mathbf{G}}\tilde{\gamma}_{ij})\mathbf{V}f(x_j)\right\|. &(58)
\end{aligned}$$

Note that $\mathbf{x}_j-f(x_j)$ represents the output difference between a GNN and an MNN evaluated at node x_j. By the transferable-PE hypothesis of Theorem 1, this difference is bounded by $\Delta_{\mathbf{\Psi}(\mathbf{G})}$. Using this bound, we can control the first term of (58) as follows:

$$\begin{aligned}
\frac{1}{D_{\mathbf{G}}}\left\|\sum_{j=1}^{N}\gamma_{ij}\mathbf{V}(\mathbf{x}_j-f(x_j))\right\|&\leq\frac{1}{D_{\mathbf{G}}}\sum_{j=1}^{N}\gamma_{ij}\|\mathbf{V}\|\cdot\|\mathbf{x}_j-f(x_j)\|\leq\frac{C_V\Delta_{\mathbf{\Psi}(\mathbf{G})}}{D_{\mathbf{G}}}\sum_{j=1}^{N}\gamma_{ij} &(59)\\
&=C_V\Delta_{\mathbf{\Psi}(\mathbf{G})}, &(60)
\end{aligned}$$

where we first apply the triangle inequality, along with the definition of the operator norm. Finally (60) uses the bound on the operator norm of $\mathbf{V}$ and $\|\mathbf{x}_j-f(x_j)\|\leq\Delta_{\mathbf{\Psi}(\mathbf{G})}$ from the transferable-PE hypothesis of Theorem 1, followed by the identity $\sum_{j=1}^{N}\gamma_{ij}=D_{\mathbf{G}}$.

Now we bound the second term in (58). We add and subtract another comparative term $\bar{D}_{\mathcal{M}}\tilde{\gamma}_{ij}$,

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

$$= \left\| \frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \sum_{j=1}^{N} \Big[(\bar{D}_{\mathcal{M}}\gamma_{ij} - \bar{D}_{\mathcal{M}}\tilde{\gamma}_{ij}) + (\bar{D}_{\mathcal{M}}\tilde{\gamma}_{ij} - D_{\mathbf{G}}\tilde{\gamma}_{ij}) \Big] \mathbf{V}f(x_j) \right\| \tag{61}$$

$$= \left\| \frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \sum_{j=1}^{N} \Big[\bar{D}_{\mathcal{M}}(\gamma_{ij} - \tilde{\gamma}_{ij}) + (\bar{D}_{\mathcal{M}} - D_{\mathbf{G}})\tilde{\gamma}_{ij} \Big] \mathbf{V}f(x_j) \right\| \tag{62}$$

$$\leq \sum_{j=1}^{N} \left\| \frac{\bar{D}_{\mathcal{M}}}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \left[(\gamma_{ij} - \tilde{\gamma}_{ij}) \right] \mathbf{V}f(x_j) \right\| + \left\| \frac{\tilde{\gamma}_{ij}(\bar{D}_{\mathcal{M}} - D_{\mathbf{G}})}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \mathbf{V}f(x_j) \right\| \tag{63}$$

$$\leq \sum_{j=1}^{N} \|\mathbf{V}\| \|f(x_j)\| \left[\frac{1}{D_{\mathbf{G}}} |\gamma_{ij} - \tilde{\gamma}_{ij}| + \frac{\tilde{\gamma}_{ij}}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} |\bar{D}_{\mathcal{M}} - D_{\mathbf{G}}| \right] \tag{64}$$

$$\leq C_V \sum_{j=1}^{N} \left[\frac{1}{D_{\mathbf{G}}} |\gamma_{ij} - \tilde{\gamma}_{ij}| + \frac{\tilde{\gamma}_{ij}}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} |\bar{D}_{\mathcal{M}} - D_{\mathbf{G}}| \right] \tag{65}$$

$$\leq C_V \left[\frac{N \cdot 2e^{B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})}}{D_{\mathbf{G}}} + \frac{|\bar{D}_{\mathcal{M}} - D_{\mathbf{G}}|}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \sum_{j=1}^{N} \tilde{\gamma}_{ij} \right] \tag{66}$$

$$\leq C_V \left[2e^{2B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})} + \frac{|\bar{D}_{\mathcal{M}} - D_{\mathbf{G}}|}{D_{\mathbf{G}}} \right] \tag{67}$$

On (62) we rearrange the terms, on (63) we apply the triangle inequality, on (64) we use the operator norm inequality, and on (65) we apply Assumptions 1 and 2. On (66) we apply the bound from Lemma 9. Finally, (67) uses $\sum_{j=1}^{N} \tilde{\gamma}_{ij} = \bar{D}_{\mathcal{M}}$ for the second term, and for the first term we note that the worst case denominator is $D_{\mathbf{G}} \geq N \cdot e^{-B}$, which we can rearrange to bound $N/D_{\mathbf{G}} \leq e^{B}$.

We can bound the remaining term using Lemma 9:

$$\frac{|\bar{D}_{\mathcal{M}} - D_{\mathbf{G}}|}{D_{\mathbf{G}}} = \frac{1}{D_{\mathbf{G}}} \left| \sum_{j=1}^{N} \tilde{\gamma}_{ij} - \sum_{j=1}^{N} \gamma_{ij} \right| \tag{68}$$

$$\leq \frac{1}{D_{\mathbf{G}}} \sum_{j=1}^{N} |\tilde{\gamma}_{ij} - \gamma_{ij}| \tag{69}$$

$$\leq \frac{N}{D_{\mathbf{G}}} \cdot 2e^{B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})} \tag{70}$$

$$\leq 2e^{2B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})}, \tag{71}$$

where in (69) we apply the triangle inequality, in (70) we apply Lemma 9, and in (71) we again use $N/D_{\mathbf{G}} \leq e^{B}$.

Substituting (71) into (67), the second term of (58) is bounded by

$$\left\| \frac{1}{D_{\mathbf{G}}\bar{D}_{\mathcal{M}}} \sum_{j=1}^{N} (\bar{D}_{\mathcal{M}}\gamma_{ij} - D_{\mathbf{G}}\tilde{\gamma}_{ij})\mathbf{V}f(x_j) \right\| \leq C_V \left[2e^{2B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})} + 2e^{2B} C_{QK} \Delta_{\mathbf{\Psi}(\mathbf{G})} \right] \tag{72}$$

$$= 4e^{2B} C_{\mathcal{T}} \Delta_{\mathbf{\Psi}(\mathbf{G})}, \tag{73}$$

where we absorb the linear operator constants into $C_{\mathcal{T}} = C_{QK} C_V$. Combining (60) and (73), we conclude the bound for the first term of (54) is

$$\left\| \mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) \right\| \leq C_V \Delta_{\mathbf{\Psi}(\mathbf{G})} + 4e^{2B} C_{\mathcal{T}} \Delta_{\mathbf{\Psi}(\mathbf{G})}. \tag{74}$$

We now bound the second term of (54). Let $\tilde{D}_{\mathcal{M}} = \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} d\mu(y)$ denote the discrete approximation of $D_{\mathcal{M}}$ in integral form,

$$
\begin{aligned}
&\|\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \\
&= \left\| \frac{1}{\tilde{D}_{\mathcal{M}}} \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y) - \frac{1}{D_{\mathcal{M}}} \int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \right\| && (75) \\
&= \left\| \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \left(D_{\mathcal{M}} \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y) - \tilde{D}_{\mathcal{M}} \int_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \right) \right\| && (76) \\
&= \left\| \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \left(D_{\mathcal{M}} \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) d\mu(y) - \tilde{D}_{\mathcal{M}} \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{iy} \mathbf{V} f(y) d\mu(y) \right) \right\| && (77) \\
&= \left\| \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \sum_{j=1}^{N} \int_{V_j} \left(D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) \right) d\mu(y) \right\| && (78) \\
&\leq \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \sum_{j=1}^{N} \int_{V_j} \left\| D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) \right\| d\mu(y) && (79)
\end{aligned}
$$

In (76) we distribute the denominators, in (77) we partition the integral, in (78) we rearrange terms, and in (79) we apply the triangle inequality.

To bound the integrand of (79), we add and subtract $D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(y)$ and apply the triangle inequality:

$$
\begin{aligned}
&\left\| D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) \right\| \\
&\leq \|D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(x_j) - D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(y)\| + \left\| D_{\mathcal{M}} \tilde{\gamma}_{ij} \mathbf{V} f(y) - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \mathbf{V} f(y) \right\| && (80) \\
&\leq D_{\mathcal{M}} \|\mathbf{V}(f(x_j) - f(y))\| + \left\| (D_{\mathcal{M}} \tilde{\gamma}_{ij} - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy}) \mathbf{V} f(y) \right\|. && (81)
\end{aligned}
$$

The first term of (81) is bounded by

$$
\begin{aligned}
D_{\mathcal{M}} \|\mathbf{V}(f(x_j) - f(y))\| &\leq D_{\mathcal{M}} \|\mathbf{V}\| \|f(x_j) - f(y)\| && (82) \\
&\leq D_{\mathcal{M}} C_V \|f(x_j) - f(y)\| && (83) \\
&\leq D_{\mathcal{M}} C_V r. && (84)
\end{aligned}
$$

In (82) we use linear operator bound, in (83) we use Assumption 1, and in (84) we use the fact that $\|f(x_j) - f(y)\| \leq r$ for all $y \in V_j$ and Assumption 2 (f is normalized Lipschitz continuous).

The second term of (81) is bounded by

$$
\begin{aligned}
\left\| (D_{\mathcal{M}} \tilde{\gamma}_{ij} - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy}) \mathbf{V} f(y) \right\| &\leq \left| D_{\mathcal{M}} \tilde{\gamma}_{ij} - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \right| \|\mathbf{V}\| \|f(y)\| && (85) \\
&\leq C_V \left| D_{\mathcal{M}} \tilde{\gamma}_{ij} - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \right| && (86) \\
&= C_V \left| D_{\mathcal{M}} \tilde{\gamma}_{ij} - D_{\mathcal{M}} \tilde{\gamma}_{iy} + D_{\mathcal{M}} \tilde{\gamma}_{iy} - \tilde{D}_{\mathcal{M}} \tilde{\gamma}_{iy} \right| && (87) \\
&\leq C_V \left(D_{\mathcal{M}} \left| \tilde{\gamma}_{ij} - \tilde{\gamma}_{iy} \right| + \left| D_{\mathcal{M}} - \tilde{D}_{\mathcal{M}} \right| \tilde{\gamma}_{iy} \right) && (88)
\end{aligned}
$$

In (85) we apply the linear operator bound, in (86) we use Assumption 1 ($\|\mathbf{V}\| \leq C_V$) and $\|f(y)\| \leq 1$ (Assumption 2), in (87) we add and subtract $D_{\mathcal{M}} \tilde{\gamma}_{iy}$, and in (88) we apply the triangle inequality and factor.

To bound the difference between denominator terms:

$$\left|D_{\mathcal{M}} - \tilde{D}_{\mathcal{M}}\right| = \left|\int_{\mathcal{M}} \tilde{\gamma}_{iy} d\mu(y) - \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{ij} d\mu(y)\right| \tag{89}$$

$$= \left|\sum_{j=1}^{N} \int_{V_j} (\tilde{\gamma}_{iy} - \tilde{\gamma}_{ij}) d\mu(y)\right| \tag{90}$$

$$\leq \sum_{j=1}^{N} \int_{V_j} |\tilde{\gamma}_{iy} - \tilde{\gamma}_{ij}| \, d\mu(y) \tag{91}$$

$$\leq \sum_{j=1}^{N} \int_{V_j} e^B C_{QK} r \, d\mu(y) \tag{92}$$

$$= e^B C_{QK} r, \tag{93}$$

where in (90) we partition the integral, in (91) we apply the triangle inequality, and in (92) we apply Lemma 8 and the fact that $\mu(\mathcal{M}) = 1$.

Now, applying (93) and Lemma (8) to (88) we have that the second term of (81) is bounded by

$$\left\|(D_{\mathcal{M}}\tilde{\gamma}_{ij} - \tilde{D}_{\mathcal{M}}\tilde{\gamma}_{iy})\mathbf{V}f(y)\right\| \leq C_V \left(D_{\mathcal{M}} e^B C_Q C_K \, r + e^B C_{QK} r \tilde{\gamma}_{iy}\right) \tag{94}$$

$$\leq D_{\mathcal{M}} e^B C_{\mathcal{T}} r + e^B C_{\mathcal{T}} r \tilde{\gamma}_{iy} \tag{95}$$

Now, we return to bound the integral of (79),

$$\|\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \tag{96}$$

$$\leq \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \sum_{j=1}^{N} \int_{V_j} \left\|D_{\mathcal{M}}\tilde{\gamma}_{ij}\mathbf{V}f(x_j) - \tilde{D}_{\mathcal{M}}\tilde{\gamma}_{iy}\mathbf{V}f(y)\right\| d\mu(y) \tag{97}$$

$$\leq \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \sum_{j=1}^{N} \int_{V_j} \left[D_{\mathcal{M}} C_V r + D_{\mathcal{M}} e^B C_{\mathcal{T}} r + e^B C_{\mathcal{T}} r \tilde{\gamma}_{iy}\right] d\mu(y) \tag{98}$$

$$\leq \frac{1}{\tilde{D}_{\mathcal{M}} D_{\mathcal{M}}} \left[D_{\mathcal{M}} C_V r \mu(\mathcal{M}) + D_{\mathcal{M}} e^B C_{\mathcal{T}} r \mu(\mathcal{M}) + e^B C_{\mathcal{T}} r \sum_{j=1}^{N} \int_{V_j} \tilde{\gamma}_{iy} d\mu(y)\right] \tag{99}$$

$$= \frac{C_V r \mu(\mathcal{M})}{\tilde{D}_{\mathcal{M}}} + \frac{e^B C_{\mathcal{T}} r \mu(\mathcal{M})}{\tilde{D}_{\mathcal{M}}} + \frac{e^B C_{\mathcal{T}} r \mu(\mathcal{M})}{D_{\mathcal{M}}} \tag{100}$$

$$\leq \frac{C_V r \mu(\mathcal{M})}{e^{-B}\mu(\mathcal{M})} + \frac{e^B C_{\mathcal{T}} r \mu(\mathcal{M})}{e^{-B}\mu(\mathcal{M})} + \frac{e^B C_{\mathcal{T}} r \mu(\mathcal{M})}{e^{-B}\mu(\mathcal{M})} \tag{101}$$

$$\leq e^B C_V r + 2e^{2B} C_{\mathcal{T}} r \tag{102}$$

where in (98) we use the bounds of (84) and (95), in (99) we evaluate the integrals of the first two terms, in (101) we distribute the denominators and use the fact that both denominators are positive quantities greater than $e^{-B}\mu(\mathcal{M})$.

We conclude that the second term of (54) is bounded by

$$\|\bar{\mathbf{\Phi}}_{\mathcal{M}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \leq e^B C_V r + 2e^{2B} C_{\mathcal{T}} r \tag{103}$$

Finally, we can bound (54) using (74) and (103), by gathering terms. With this we conclude that the output difference of GT and MT is bounded by

$$\|\mathbf{\Phi}_{\mathbf{G}}(\mathcal{T}, \mathbf{X})(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i)\| \leq C_V \Delta_{\mathbf{\Psi}(\mathbf{G})} + 4e^{2B} C_{\mathcal{T}} \Delta_{\mathbf{\Psi}(\mathbf{G})} + e^B C_V r + 2e^{2B} C_{\mathcal{T}} r \tag{104}$$

$$\leq \left[C_V + 4e^{2B} C_{\mathcal{T}}\right] \Delta_{\mathbf{\Psi}(\mathbf{G})} + \left[e^B C_V + 2e^{2B} C_{\mathcal{T}}\right] r. \tag{105}$$

which gives us the statement of Theorem 1. □

C.4 Proof of Corollary 2

Corollary 2 bounds the output difference between two GTs trained with differently sized graphs by applying Theorem 1.

Proof. The output difference can be bounded as

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

$$=\frac{1}{\mu(\mathcal{M})}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)+\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\|_{L^{1,2}(\mathcal{M})} \tag{106}$$

$$\leq\frac{1}{\mu(\mathcal{M})}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)\|_{L^{1,2}(\mathcal{M})}+\frac{1}{\mu(\mathcal{M})}\|\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\|_{L^{1,2}(\mathcal{M})} \tag{107}$$

$$\leq\frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}}\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)(x)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x)\|_2\,d\mu(x) \tag{108}$$

$$+\frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}}\|\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)(x)\|_2\,d\mu(x) \tag{109}$$

where in (106) we add and subtract $\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)$, in (107) we apply the triangle inequality, and in (108) we use the definition of $L^{1,2}(\mathcal{M})$.

The two terms in (109) 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 $\mathbf{\Phi}_{\mathcal{M}}$,

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

$$=\frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N_1}\int_{V_i}\|\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)(x_i)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x)\|_2\,d\mu(x) \tag{110}$$

$$=\frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N_1}\int_{V_i}\|\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)(x_i)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x_i)+\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x_i)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x)\|_2\,d\mu(x) \tag{111}$$

$$\leq\frac{1}{\mu(\mathcal{M})}\sum_{i=1}^{N_1}\int_{V_i}\|\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)(x_i)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x_i)\|_2+\|\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x_i)-\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T},f)(x)\|_2\,d\mu(x) \tag{112}$$

In (110) we apply the interpolation operator on $\mathbf{\Phi}_{\mathbf{G}_1}$, where $\{V_i\}_{i=1}^{N_1}$ is the partition induced by $\mathbf{G}_1$, then in (112) we add and subtract the MT output at point x_i, and apply the triangle inequality. The first term of (112) is the statement of Theorem 1. 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

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

$$\leq\left\|\frac{\int_{\mathcal{M}}\tilde{\gamma}_{iy}\mathbf{V}f(y)d\mu(y)}{\int_{\mathcal{M}}\tilde{\gamma}_{iy}d\mu(y)}-\frac{\int_{\mathcal{M}}\tilde{\gamma}_{xy}\mathbf{V}f(y)d\mu(y)}{\int_{\mathcal{M}}\tilde{\gamma}_{xy}d\mu(y)}\right\| \tag{113}$$

$$\leq\left\|\frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}}\Big(D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)-D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\Big)d\mu(y)\right\| \tag{114}$$

$$\leq\frac{1}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}}\Big\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y)-D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\Big\|\,d\mu(y) \tag{115}$$

Which we obtain by distributing the denominators, applying the triangle inequality, and grouping terms. The integrand in (115) is bounded as

$$\begin{aligned}
&\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\| \\
&\leq \|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\left[\mathbf{V}f(x) - \mathbf{V}f(y)\right] - \left[D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\right]\mathbf{V}f(y)\| && (116)\\
&\leq D_{\mathcal{M}}(x)C_V\tilde{\gamma}_{iy}\|f(x) - f(y)\| + C_V\left|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\right| && (117)
\end{aligned}$$

where in (116) we add and subtract $D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(x)$, and in (117) we apply the triangle inequality and the bound on $\mathbf{V}$.

We now focus on the second term of (117),

$$\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy} - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\| \leq D_{\mathcal{M}}(x)|\tilde{\gamma}_{iy} - \tilde{\gamma}_{xy}| + \|\left[D_{\mathcal{M}}(x) - D_{\mathcal{M}}(x_i)\right]\tilde{\gamma}_{xy}\| \tag{118}$$

The second term of (118) is

$$\begin{aligned}
\tilde{\gamma}_{xy}\|D_{\mathcal{M}}(x) - D_{\mathcal{M}}(x_i)\| &= \left\|\int_{\mathcal{M}} \tilde{\gamma}_{xy}\tilde{\gamma}_{xy} - \tilde{\gamma}_{iy}d\mu(y)\right\| && (119)\\
&\leq \int_{\mathcal{M}} \tilde{\gamma}_{xy}\left\|\tilde{\gamma}_{xy} - \tilde{\gamma}_{iy}\right\| d\mu(y) && (120)\\
&\leq D_{\mathcal{M}}(x_i)e^B C_{QK} r && (121)
\end{aligned}$$

where we use Lemma 10 to bound the remaining integral by $D_{\mathcal{M}}(x_i)$.

Returning to the integral (115), we have

$$\begin{aligned}
&\frac{C_V}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} \left\|D_{\mathcal{M}}(x)\tilde{\gamma}_{iy}\mathbf{V}f(y) - D_{\mathcal{M}}(x_i)\tilde{\gamma}_{xy}\mathbf{V}f(y)\right\| d\mu(y) && (122)\\
&\quad\leq \frac{C_V}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}\int_{\mathcal{M}} \left\|D_{\mathcal{M}}(x)e^B C_{QK} r + \tilde{\gamma}_{xy}\mu(\mathcal{M})e^B C_{QK} r\right\| d\mu(y) && (123)\\
&\quad= \frac{C_V}{D_{\mathcal{M}}(x)D_{\mathcal{M}}(x_i)}D_{\mathcal{M}}(x)\mu(\mathcal{M})e^B C_{QK} r + \tilde{\gamma}_{xy}D_{\mathcal{M}}(x)\mu(\mathcal{M})e^B C_{QK} r && (124)\\
&\quad= \frac{1}{D_{\mathcal{M}}(x_i)}2\mu(\mathcal{M})e^B C_{\mathcal{T}} r && (125)\\
&\quad\leq 2e^{2B} C_{\mathcal{T}} r && (126)
\end{aligned}$$

by applying the bounds of (121) and Lemma 10, then evaluating the integral, and finally using the fact that the denominator is $D_{\mathcal{M}}(x_i) \geq e^{-B}\mu(\mathcal{M})$. We conclude that

$$\|\mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x_i) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x)\|_2 \leq 2e^{2B} C_{\mathcal{T}} r \tag{127}$$

We can now bound (108) by applying this bound and the bound of Theorem 1,

$$\frac{1}{\mu(\mathcal{M})}\int_{\mathcal{M}} \|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T}, \mathbf{X}_1)(x) - \mathbf{\Phi}_{\mathcal{M}}(\mathcal{T}, f)(x)\|_2 \, d\mu(x) \leq \Delta_{\mathbf{\Phi}(\mathbf{G}_1)} + 2e^{2B} C_{\mathcal{T}}\, r_1, \tag{128}$$

where $\Delta_{\mathbf{\Phi}(\mathbf{G}_1)}$ denotes the bound of Theorem 1 for $\mathbf{G}_1$, and r_1 its partition radius.

Applying this bound in (109) for $\mathbf{\Phi}(\mathbf{X}_1)$ and $\mathbf{\Phi}(\mathbf{X}_2)$ gives us the statement of Corollary 2.

□

D Convergence of Positional Encodings

D.1 Proof of Theorem 3

Proof. Our setting is the same as Theorem 4 in Dunson et al. (2021), we restate their conclusion in our notations as: with probability greater than $1 - N^{-2}$ there are signs $r_{j,N} \in \{-1, 1\}$, simultaneously for $1 \leq j \leq K$, such that

$$\max_{1\leq i\leq N} |r_{j,N}[\boldsymbol{\phi}_{j,N}]_i - \boldsymbol{\phi}_j(x_i)| \leq \frac{\Omega_K}{\Gamma_K}\varepsilon_N^2, \tag{129}$$

with $\Gamma_K := \min_{1\le j\le K}(\lambda_{j+1}-\lambda_j)$ and $\varepsilon_N \ge \left(\frac{\log N}{N}\right)^{1/(2d+8)}$, and Ω_K depends on λ_K, d, and the diameter, the volume, the injectivity radius, the curvature and the second fundamental form of the manifold.

Define $\mathbf{R}_N = \mathrm{diag}(r_{1,N},\dots,r_{K,N})$, for every $x_i \in X_N$,

$$\left\|\mathbf{R}_N \mathbf{\Psi}_{\mathbf{G}}^{(K)}(\mathbf{L}_N)(x_i) - \mathbf{\Psi}_{\mathcal{M}}^{(K)}(\mathcal{L})(x_i)\right\|_2^2 = \sum_{j=1}^{K} |r_{j,N}[\boldsymbol{\phi}_{j,N}]_i - \boldsymbol{\phi}_j(x_i)|^2 \tag{130}$$

$$\le K\,\Omega_K^2\,\Gamma_K^{-2}\,\varepsilon_N^4. \tag{131}$$

Taking square roots yields the bound. □

D.2 Graph and Manifold Eigengap

In Section 5.1 we discussed the dependence of the eigenvector PE convergence on the Laplace-Beltrami operator's eigengap. Here we discuss the relationship between $\Gamma_{K,N}$ and Γ_K. Using Proposition 5, we have

$$|(\lambda_{j+1,N}-\lambda_{j,N}) - (\lambda_{j+1}-\lambda_j)| \le 2\,\Omega_1\,\varepsilon_N^2, \qquad 1\le j\le K, \tag{132}$$

Thus the gaps are related by

$$|\Gamma_{K,N} - \Gamma_K| \le 2\,\Omega_1\,\varepsilon_N^2 \tag{133}$$

Hence $\Gamma_{K,N}$ is a reliable estimate of Γ_K only when the sampling error $2\Omega_1\varepsilon_N^2$ is small relative to Γ_K, which is precisely the regime $N \ge N_0(K,\delta)$ where Theorem 3 holds.

D.3 Manifold neural networks and their relation to GNNs

Laplace-Beltrami operator and manifold convolutions. The manifold with probability density function ρ is equipped with a weighted Laplace operator (Grigor'yan, 2006), generalizing the Laplace-Beltrami operator as

$$\mathcal{L}_\rho f = -\frac{1}{2\rho}\mathrm{div}(\rho^2 \nabla f), \tag{134}$$

with div denoting the divergence operator of $\mathcal{M}$ and ∇ denoting the gradient operator of $\mathcal{M}$ (Bronstein et al., 2017; Gross & Meinrenken, 2023). The manifold convolution operation is defined relying on the Laplace operator $\mathcal{L}$ (Wang et al., 2024b). For notational simplicity, we write $\mathcal{L}_\rho = \mathcal{L}$. For a function $g \in L^2(\mu)$ as input, a manifold convolutional filter (Wang et al., 2024b) can be defined as

$$f(x) = \mathbf{h}(\mathcal{L})g(x) = \sum_{k=0}^{K-1} h_k e^{-k\mathcal{L}} g(x), \tag{135}$$

with $h_k \in \mathbb{R}$ the filter parameters. The polynomial counterpart of the graph filter in Equation (2) is

$$f(x) = \mathbf{h}(\mathcal{L})g(x) = \sum_{k=0}^{K-1} h_k \mathcal{L}^k g(x). \tag{136}$$

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

$$f_l^p(x) = \sigma\left(\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x)\right), \tag{137}$$

where f_{l-1}^q, $1 \le q \le F_{l-1}$ is the q-th input feature from layer $l-1$ and f_l^p, $1 \le p \le F_l$ is the p-th output feature of layer l. The input to the first layer is the manifold signal, $f_0^q = g^q$, and the MNN output is the last layer, $f = f_L$. We denote MNN as a mapping $f = \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)$, where $\mathcal{H} = \{\mathbf{h}_l^{pq}\}_{l,p,q}$ is a collective set of filter parameters in all the manifold convolutional filters.

D.4 Proof of Theorem 4

We impose the following additional assumption for the GNN to MNN convergence proof.

Assumption 4 (Normalized Lipschitz signals)**.** *The input manifold signals g are normalized Lipschitz for all points $a, b \in \mathcal{M}$, $\|g(b) - g(a)\| \leq \|b - a\|$.*

Assumption 5 (Normalized Lipschitz nonlinearity)**.** *The nonlinearity σ is normalized Lipschitz continuous, i.e.,*

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

Assumption 6 (Non-amplifying filters)**.** *The filter frequency response $\hat{h} : \mathbb{R}^+ \to \mathbb{R}$ is non-amplifying. I.e., for all $\lambda \in (0, \infty)$, $\hat{h}$ satisfies $|\hat{h}(\lambda)| \leq 1$.*

Assumption 7 (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.

Assumptions 4–7 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. Building on this foundation, we present a convergence theorem from the literature, tailored to our setting.

Theorem 4 provides the bound $\Delta_{\mathbf{\Psi}(\mathbf{G})}$ on the output difference between a GNN and its MNN limit required by the transferable-PE hypothesis of Theorem 1; its proof follows.

Proof of Theorem 4. 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, $\mathcal{L}$ has real, nonnegative and discrete eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$, written as $\mathcal{L}\boldsymbol{\phi}_i = \lambda_i \boldsymbol{\phi}_i$ 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 $L^2(\mu)$. When mapping a manifold signal onto the eigenbasis $[\hat{g}]_i = \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} = \int_{\mathcal{M}} g(x)\boldsymbol{\phi}_i(x)\mathrm{d}\mu(x)$, the manifold convolution can be seen in the spectral domain as

$$[\hat{f}]_i = \hat{h}(\lambda_i)[\hat{g}]_i. \tag{138}$$

Hence, the frequency response of manifold filter is given by $\hat{h}(\lambda)$. We will use the spectral convergence of the graph Laplacian to the Laplace–Beltrami operator stated in Proposition 5 (Appendix B.2).

Because $\{x_1, x_2, \cdots, x_N\}$ is a set of randomly sampled points from $\mathcal{M}$, based on Theorem 19 in (Von Luxburg et al., 2008) we can claim that

$$|\langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}}| = O\left(\sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{139}$$

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

$$\left| \|\mathbf{S}_N g\|^2 - \|g\|_{\mathcal{M}}^2 \right| = O\left(\sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{140}$$

which indicates $\|\mathbf{S}_N g\| = \|g\|_{\mathcal{M}} + O((\log(1/\delta)/N)^{1/4})$, where $\|g\|_{\mathcal{M}}^2 = \langle g, g \rangle_{\mathcal{M}}$. We suppose that the input manifold signal is λ_J-bandlimited with J spectral components. We first write out the

difference on each node $x_j \in X_N$ as

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{S}_N g]_j - (\mathbf{h}(\mathcal{L})g)(x_j)\| = \left\| \sum_{i=1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{J} \hat{h}(\lambda_i) \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| \tag{141}$$

$$\leq \left\| \sum_{i=1}^{J} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{J} \hat{h}(\lambda_i) \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) + \sum_{i=J+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \tag{142}$$

$$\leq \left\| \sum_{i=1}^{J} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{J} \hat{h}(\lambda_i) \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| + \left\| \sum_{i=J+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\|. \tag{143}$$

The first part of (143) can be decomposed with the triangle inequality as follows

$$\begin{aligned} &\left\| \sum_{i=1}^{J} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \sum_{i=1}^{J} \hat{h}(\lambda_i) \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right\| \\ &\leq \left\| \sum_{i=1}^{J} \left(\hat{h}(\lambda_{i,N}) - \hat{h}(\lambda_i) \right) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| + \left\| \sum_{i=1}^{J} \hat{h}(\lambda_i) \left(\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\|. \end{aligned} \tag{144}$$

In (144), the first part relies on the difference of eigenvalues and the second part depends on the eigenvector difference. The first term in (144) is bounded with Cauchy-Schwartz inequality as

$$\left\| \sum_{i=1}^{J} (\hat{h}(\lambda_{i,N}) - \hat{h}(\lambda_i)) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \leq \sum_{i=1}^{J} \left| \hat{h}(\lambda_{i,N}) - \hat{h}(\lambda_i) \right| |\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle| \tag{145}$$

$$\leq \|\mathbf{S}_N g\| \sum_{i=1}^{J} |\hat{h}'(\lambda_i)| |\lambda_{i,N} - \lambda_i|$$

$$\leq \|\mathbf{S}_N g\| \sum_{i=1}^{J} C_L \Omega_1 \epsilon^2 \lambda_i^{-d} \tag{146}$$

$$\leq \|\mathbf{S}_N g\| C_L \Omega_1 \epsilon^2 \sum_{i=1}^{J} i^{-2} \tag{147}$$

$$\leq \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) C_L \Omega_1 \epsilon^2 \frac{\pi^2}{6} := A_1(N) \tag{148}$$

In (145), 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 (146), it depends on the filter continuity assumption. In (147), we implement Weyl's law (Arendt et al., 2009) 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 (144) can be bounded with the triangle

inequality as

$$\begin{aligned}
&\left\| \sum_{i=1}^{J} \hat{h}(\lambda_i) \left(\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\leq \left\| \sum_{i=1}^{J} \hat{h}(\lambda_i) \left(\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\qquad + \left\| \sum_{i=1}^{J} \hat{h}(\lambda_i) \left(\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j) \right) \right\| \qquad (149)
\end{aligned}$$

The first term in (149) can be bounded with inserting the eigenfunction convergence result in Proposition 5 as

$$\begin{aligned}
&\left\| \sum_{i=1}^{J} \hat{h}(\lambda_i) \left(\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j - \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) \right) \right\| \\
&\leq \sum_{i=1}^{J} \left| \hat{h}(\lambda_i) \right| \| \mathbf{S}_N g \| \left| [\boldsymbol{\phi}_{i,N}]_j - \boldsymbol{\phi}_i(x_j) \right| \qquad (150) \\
&\leq \sum_{i=1}^{J} (\lambda_i^{-d+1}) \Omega_2 \epsilon^2 \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \qquad (151) \\
&\leq \Omega_2 \epsilon^2 \sum_{i=1}^{J} (\lambda_i^{-d+1}) \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \qquad (152) \\
&:= A_2(J, N). \qquad (153)
\end{aligned}$$

Considering the filter continuity assumption, the second term in (149) can be written as

$$\begin{aligned}
&\left\| \sum_{i=1}^{J} \hat{h}(\lambda_{i,N}) (\langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle \boldsymbol{\phi}_i(x_j) - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \boldsymbol{\phi}_i(x_j)) \right\| \\
&\leq \sum_{i=1}^{J} \left| \hat{h}(\lambda_{i,N}) \right| \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| |\boldsymbol{\phi}_i(x_j)| \qquad (154) \\
&\leq \sum_{i=1}^{J} (\lambda_{i,N}^{-d}) \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \|\boldsymbol{\phi}_i\| \qquad (155) \\
&\leq \sum_{i=1}^{J} \left(1 + \Omega_1 \epsilon^2 \right)^{-d} \lambda_i^{-d} \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (156) \\
&\leq \frac{\pi^2}{6} \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| := A_3(N) \qquad (157)
\end{aligned}$$

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

$$\begin{aligned}
\left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| &\leq \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle + \langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (158) \\
&\leq \left| \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle - \langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle \right| + \left| \langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (159) \\
&\leq \|\mathbf{S}_N g\| \|\boldsymbol{\phi}_{i,N} - \mathbf{S}_N \boldsymbol{\phi}_i\| + \left| \langle \mathbf{S}_N g, \mathbf{S}_N \boldsymbol{\phi}_i \rangle - \langle g, \boldsymbol{\phi}_i \rangle_{\mathcal{M}} \right| \qquad (160) \\
&\leq \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \frac{C_{\mathcal{M},2} \lambda_i \epsilon}{\theta_i} + \sqrt{\frac{\log(1/\delta)}{N}} \qquad (161)
\end{aligned}$$

Then equation (156) can be bounded as

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

$$\leq \sum_{i=1}^{J} (1+\Omega_1 \epsilon^2)^{-d} (\lambda_i^{-d}) \left(\left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \frac{C_{\mathcal{M},2} \lambda_i \epsilon}{\theta_i} + \sqrt{\frac{\log(1/\delta)}{N}} \right) \tag{162}$$

$$\leq \frac{\pi^2}{6} \max_{i=1,\cdots,J} \frac{C_{\mathcal{M},2}\epsilon}{\theta_i} \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) + \frac{\pi^2}{6} \sqrt{\frac{\log(1/\delta)}{N}} \tag{163}$$

The second term in (143) can be bounded with the eigenvalue difference bound in Proposition 5 as

$$\left\| \sum_{i=J+1}^{N} \hat{h}(\lambda_{i,N}) \langle \mathbf{S}_N g, \boldsymbol{\phi}_{i,N} \rangle [\boldsymbol{\phi}_{i,N}]_j \right\| \leq \sum_{i=J+1}^{N} (\lambda_{i,N}^{-d}) \left(\|g\|_{\mathcal{M}} + \left(\frac{\log(1/\delta)}{N} \right)^{\frac{1}{4}} \right) \tag{164}$$

$$\leq \sum_{i=J+1}^{\infty} (\lambda_{i,N}^{-d}) \|g\|_{\mathcal{M}} \tag{165}$$

$$\leq \left(1+\Omega_1 \epsilon^2\right)^{-d} \sum_{i=J+1}^{\infty} (\lambda_i^{-d}) \|g\|_{\mathcal{M}} \tag{166}$$

$$\leq J^{-1} \|g\|_{\mathcal{M}} := A_4(J). \tag{167}$$

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

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{S}_N g]_j - \mathbf{h}(\mathcal{L}) g(x_j)\| \leq C_1' \epsilon^2 + C_2' \epsilon \theta_J^{-1} + C_3' \sqrt{\frac{\log(1/\delta)}{N}} + C_4' J^{-1}, \tag{168}$$

with $C_1' = C_L \Omega_1 \frac{\pi^2}{6} \|g\|_{\mathcal{M}}$, $C_2' = \Omega_2 \frac{\pi^2}{6}$, $C_3' = \frac{\pi^2}{6}$ and $C_4' = \|g\|_{\mathcal{M}}$. As N goes to infinity, for every $\delta > 0$, there exists some J_0, such that for all $J > J_0$ it holds that $A_4(J) \leq \delta/2$. There also exists n_0, such that for all $N > n_0$, it holds that $A_1(N) + A_2(J_0, N) + A_3(N) \leq \delta/2$. We can conclude that the summations converge as N goes to infinity. We see J large enough to have $J^{-1} \leq \delta'$, which makes the eigengap θ_J also bounded by ϵ. We combine the first two terms as

$$\|[\mathbf{h}(\mathbf{L}_N)\mathbf{S}_N g]_j - \mathbf{h}(\mathcal{L}) g(x_j)\| \leq (C_1 C_L + C_2)\epsilon^2 + \frac{\pi^2}{6} \sqrt{\frac{\log(1/\delta)}{N}}, \tag{169}$$

with $C_1 = \Omega_1 \frac{\pi^2}{6} \|g\|_{\mathcal{M}}$ 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

$$\|[\boldsymbol{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)]_j - \boldsymbol{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x_j)\| = \left\| \sum_{q=1}^{F} [\mathbf{x}_L^q]_j - \sum_{q=1}^{F} f_L^q(x_j) \right\| \tag{170}$$

$$\leq \sum_{q=1}^{F} \|[\mathbf{x}_L^q]_j - f_L^q(x_j)\|. \tag{171}$$

By inserting the definitions, we have

$$\|[\mathbf{x}_l^p]_j - f_l^p(x_j)\| = \left\| \sigma \left(\left[\sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{l-1}^q \right]_j \right) - \sigma \left(\sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x_j) \right) \right\| \tag{172}$$

with $\mathbf{x}_0 = \mathbf{S}_N g$ as the input of the first layer. With a normalized point-wise Lipschitz nonlinearity, we have

$$\|[\mathbf{x}_l^p]_j - f_l^p(x_j)\| \leq \left\| \sum_{q=1}^{F} \left[\mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{l-1}^q \right]_j - \sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x_j) \right\| \tag{173}$$

$$\leq \sum_{q=1}^{F} \left\| [\mathbf{h}_l^{pq}(\mathbf{L}_N) \mathbf{x}_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x_j) \right\| \tag{174}$$

The difference can be further decomposed as

$$\begin{aligned}
&\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L})f_{l-1}^q(x_j)\| \\
&\leq \|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{l-1}^q]_j - [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{S}_N f_{l-1}^q]_j + [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{S}_N f_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L})f_{l-1}^q(x_j)\| \quad (175)\\
&\leq \left\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{l-1}^q]_j - [\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{S}_N f_{l-1}^q]_j\right\| + \left\|[\mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{S}_N f_{l-1}^q]_j - \mathbf{h}_l^{pq}(\mathcal{L})f_{l-1}^q(x_j)\right\| \quad (176)
\end{aligned}$$

The second term can be bounded with (168) 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

$$\|[\mathbf{x}_l^p]_j - f_l^p(x_j)\| \leq \sum_{q=1}^{F} \Delta_N \|\mathbf{x}_{l-1}^q\| + \sum_{q=1}^{F} \|[\mathbf{x}_{l-1}^q]_j - f_{l-1}^q(x_j)\|. \quad (177)$$

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

$$\|\mathbf{x}_l^p\| \leq \left\|\sum_{q=1}^{F} \mathbf{h}_l^{pq}(\mathbf{L}_N)\mathbf{x}_{l-1}^q\right\| \leq \sum_{q=1}^{F} \|\mathbf{h}_l^{pq}(\mathbf{L}_N)\| \, \|\mathbf{x}_{l-1}^q\| \leq \sum_{q=1}^{F} \|\mathbf{x}_{l-1}^q\| \leq F^{l-1}\|\mathbf{x}\|. \quad (178)$$

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

$$\|[\mathbf{x}_l^p]_j - f_l^p(x_j)\| \leq lF^{l-1}\Delta_N \|\mathbf{x}\|. \quad (179)$$

Replace l with L we can obtain

$$\|[\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H}, \mathbf{L}_N, \mathbf{S}_N g)]_j - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x_j)\| \leq LF^{L-1}\Delta_N, \quad (180)$$

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

D.5 Local Lipschitz continuity of MNNs

We utilize Proposition 3 in (Wang et al., 2025), which shows that the outputs of MNN defined in (137) are locally Lipschitz continuous within a certain area, which is stated explicitly as follows.

Proposition 11. *(Local Lipschitz continuity of MNNs (Wang et al., 2025)[Proposition 3]) Assume that Assumptions 4–7 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*

$$\|\mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(x) - \mathbf{\Psi}_{\mathcal{M}}(\mathcal{H}, \mathcal{L}, g)(y)\| \leq F^L C' \, d_{\mathcal{M}}(x, y), \quad \textit{for all } x, y \in B_r(\mathcal{M}), \quad (181)$$

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

D.6 RPEARL Positional Encodings

In the experiments of Section 6 we train RPEARL positional encodings, which consist of a GNN processing random i.i.d. features. In this section we define the architecture explicitly. A single pass of a GNN is given by:

$$\mathbf{P} = \sigma\left[\sum_{k=0}^{K-1} \mathbf{H}_k \mathbf{Z}\mathbf{L}^k\right] \quad (182)$$

where $\mathbf{Z} \in \mathbb{R}^{q \times N}$ are the input signals, and $\mathcal{H} = \{\mathbf{H}_0, \ldots, \mathbf{H}_{K-1}\}$ collects the filter coefficients. Equation (182) performs a graph convolution of order K on the graph signals by the graph Laplacian $\mathbf{L}$, followed by the pointwise nonlinearity σ.

Consider taking as input single-row matrices of random node features, $\mathbf{Z} = [\mathbf{z}^{(m)}]^\top$, where $\mathbf{z}^{(m)}$ is sampled from an N-dimensional isotropic Gaussian. Random features play the role of node IDs, which are known to improve the expressiveness of GNNs by breaking their structural symmetries (You et al., 2021). However, the resulting output for a single pass loses permutation equivariance. We can

recover this property by drawing M realizations $\{\mathbf{z}^{(m)}\}_{m=1}^{M}$ and taking the empirical average over independent passes of the GNN,

$$\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H},\mathbf{L},\mathbf{Z}) = \frac{1}{M}\sum_{m=1}^{M}\mathbf{P}^{(m)}, \qquad \mathbf{P}^{(m)} = \sigma\left[\sum_{k=0}^{K-1}\mathbf{H}_k\mathbf{z}^{(m)\top}\mathbf{L}^k\right], \tag{183}$$

where in this case the filters are $\mathbf{H}_k \in \mathbb{R}^{q\times 1}$. The resulting PE $\mathbf{\Psi}_{\mathbf{G}}(\mathcal{H},\mathbf{L},\mathbf{Z})$, named RPEARL, was shown by (Kanatsoulis et al., 2024) to provably enhance the expressive power of GNNs beyond the WL-test while preserving permutation equivariance and the stability and scalability of GNNs (Ruiz et al., 2021). We consider RPEARL positional encodings in our experiments as an empirical enhancement for expressivity.

E Transferability Corollaries

E.1 GT transferability corollaries

Corollary 12 (Transferability of GT with eigenvector PE)**.** *Let $\mathbf{G}_1, \mathbf{G}_2$ satisfy the assumptions of Theorems 1 and 3, and let the positional encodings be the sign-aligned eigenvectors $\mathbf{R}_{N_1}\mathbf{\Psi}^{(K)}_{\mathbf{G}_1}$ and $\mathbf{R}_{N_2}\mathbf{\Psi}^{(K)}_{\mathbf{G}_2}$, rescaled by a constant $c_K \leq 1$ depending on K so that Assumption 2 holds (the rescaling does not increase the PE error of Theorem 3). Then, with probability at least $1-\delta-N_1^{-2}-N_2^{-2}$,*

$$\begin{aligned}\frac{1}{\mu(\mathcal{M})}&\left\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\right\|_{L^{1,2}(\mathcal{M})}\\ &\leq C_{\mathbf{\Phi},1}\frac{\sqrt{K}\,\Omega_K}{\Gamma_K}\left(\varepsilon^2_{N_1}+\varepsilon^2_{N_2}\right)\\ &\quad + C_{\mathbf{\Phi},2}(r_1+r_2),\end{aligned} \tag{184}$$

where $C_{\mathbf{\Phi},1} := C_V + 4e^{2B}C_{\mathcal{T}}$ and $C_{\mathbf{\Phi},2} := e^B C_V + 6e^{2B}C_{\mathcal{T}}$.

Corollary 13 (Transferability of GT with GNN PEs)**.** *Let $\mathbf{G}_1, \mathbf{G}_2$ satisfy the assumptions of Theorems 1 and 4, with the same GNN parameters $\mathcal{H}$ and input manifold signal g. Then, with probability at least $1-\delta$,*

$$\begin{aligned}\frac{1}{\mu(\mathcal{M})}&\left\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1) - \mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\right\|_{L^{1,2}(\mathcal{M})}\\ &\leq C_{\mathbf{\Phi},1}\left[C\left(\varepsilon^2_{N_1}+\varepsilon^2_{N_2}\right) + \frac{\pi^2}{6}\left(\sqrt{\frac{\log(2/\delta)}{N_1}}+\sqrt{\frac{\log(2/\delta)}{N_2}}\right)\right]\\ &\quad + C_{\mathbf{\Phi},2}(r_1+r_2),\end{aligned} \tag{185}$$

with $C_{\mathbf{\Phi},1}, C_{\mathbf{\Phi},2}$ defined as in Corollary 12.

Contrasting the bounds in Corollaries 12 and 13 makes evident the tradeoffs of each positional encoding in the transferability of the GT. Both bounds depend on the Lipschitz constants $C_{\mathbf{\Phi},1}$ and $C_{\mathbf{\Phi},2}$ which we can control via regularization. Corollary 12 also depends on the manifold eigengap Γ_K, which, as we saw in Section 5.1, we do not have access to. In contrast, in Corollary 13 we obtain the GNN constant C, which can also be controlled via regularization.

E.2 Proof of Corollary 12

Proof. By Theorem 3, the PEs are bounded by

$$\|[\mathbf{X}_k]_i - f(x_i)\|_2 \leq \Delta_{\mathbf{\Psi}(\mathbf{G}_k)} = \frac{\sqrt{K}\,\Omega_K}{\Gamma_K}\,\varepsilon^2_{N_k}. \tag{186}$$

The proof is a straightforward application into Corollary 2,

$$\begin{aligned}
&\frac{1}{\mu(\mathcal{M})}\left\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\right\|_{L^{1,2}(\mathcal{M})}\\
&\quad\le \Delta_{\mathbf{\Phi}(\mathbf{G}_1)}+\Delta_{\mathbf{\Phi}(\mathbf{G}_2)}+4e^{2B}C_{\mathcal{T}}(r_1+r_2) &(187)\\
&\quad= \left[C_V+4e^{2B}C_{\mathcal{T}}\right]\left(\Delta_{\mathbf{\Psi}(\mathbf{G}_1)}+\Delta_{\mathbf{\Psi}(\mathbf{G}_2)}\right)\\
&\qquad\quad+\left[e^{B}C_V+2e^{2B}C_{\mathcal{T}}+4e^{2B}C_{\mathcal{T}}\right](r_1+r_2) &(188)\\
&\quad= C_{\mathbf{\Phi},1}\left(\Delta_{\mathbf{\Psi}(\mathbf{G}_1)}+\Delta_{\mathbf{\Psi}(\mathbf{G}_2)}\right)+C_{\mathbf{\Phi},2}(r_1+r_2), &(189)
\end{aligned}$$

where in (187) we apply Corollary 2, in (188) we substitute $\Delta_{\mathbf{\Phi}(\mathbf{G}_1)}$ and $\Delta_{\mathbf{\Phi}(\mathbf{G}_2)}$ from Theorem 1, and in (189) we group terms and substitute $C_{\mathbf{\Phi},1}=C_V+4e^{2B}C_{\mathcal{T}}$ and $C_{\mathbf{\Phi},2}=e^{B}C_V+6e^{2B}C_{\mathcal{T}}$.

Substituting the eigenvector PE bounds into (189),

$$\begin{aligned}
&\frac{1}{\mu(\mathcal{M})}\left\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\right\|_{L^{1,2}(\mathcal{M})}\\
&\quad\le C_{\mathbf{\Phi},1}\frac{\sqrt{K}\,\Omega_K}{\Gamma_K}\left(\varepsilon^2_{N_1}+\varepsilon^2_{N_2}\right)+C_{\mathbf{\Phi},2}(r_1+r_2), &(190)
\end{aligned}$$

where in (190) we apply Theorem 3 twice and group terms. □

E.3 Proof of Corollary 13

Proof. Similarl to the above. By Theorem 4, choosing probability $\delta/2$, GNN PEs are bounded as

$$\|[\mathbf{X}_k]_i-f(x_i)\|_2\le\Delta_{\mathbf{\Psi}(\mathbf{G}_k)}=C\varepsilon^2_{N_k}+\frac{\pi^2}{6}\sqrt{\frac{\log(2/\delta)}{N_k}}. \tag{191}$$

Substituting the GNN PE bounds into (189),

$$\begin{aligned}
&\frac{1}{\mu(\mathcal{M})}\left\|\mathbf{I}_{N_1}\mathbf{\Phi}_{\mathbf{G}_1}(\mathcal{T},\mathbf{X}_1)-\mathbf{I}_{N_2}\mathbf{\Phi}_{\mathbf{G}_2}(\mathcal{T},\mathbf{X}_2)\right\|_{L^{1,2}(\mathcal{M})}\\
&\quad\le C_{\mathbf{\Phi},1}\left[C(\varepsilon^2_{N_1}+\varepsilon^2_{N_2})+\frac{\pi^2}{6}\left(\sqrt{\frac{\log(2/\delta)}{N_1}}+\sqrt{\frac{\log(2/\delta)}{N_2}}\right)\right]\\
&\qquad+C_{\mathbf{\Phi},2}(r_1+r_2), &(192)
\end{aligned}$$

where in (192) we apply (191) twice with confidence parameter $\delta/2$ and group terms. □

E.4 Sparse Graph Transformer

One common computational tradeoff in Graph Transformers is to mask coefficients to attend to a neighborhood of nodes reachable in k-hops instead of computing all pairwise node attentions. Restricting the attention operation to the k-hop neighborhood results in the Sparse Graph Transformer (Sparse GT) given by

$$\dot{\mathbf{\Phi}}_{\mathbf{G}}(\mathcal{T},\mathbf{X})(x_i)=\frac{\sum_{j\in\mathcal{N}^{\le k}(i)}\exp\{\langle\mathbf{Q}\mathbf{x}_i,\mathbf{K}\mathbf{x}_j\rangle\}\mathbf{V}\mathbf{x}_j}{\sum_{j\in\mathcal{N}^{\le k}(i)}\exp\{\langle\mathbf{Q}\mathbf{x}_i,\mathbf{K}\mathbf{x}_j\rangle\}}, \tag{193}$$

where we denote the k-hop neighborhood of node i as $\mathcal{N}^{\le k}(i)=\bigcup_{k'=0}^{k}\mathcal{N}^{k'}(i)$, with $\mathcal{N}^0(i)=\{i\}$ and $\mathcal{N}^k(i)=\{j:\exists j'\in\mathcal{N}^{k-1}(i),\ j\in\mathcal{N}(j')\}$ for $k\ge 1$. Sparse GT reduces computational complexity from quadratic in N to quadratic in the worst-case k-hop neighborhood cardinality.

F Experimental Details

This section contains an extended discussion of the experimental methodology for the transferability experiment in the main body.

F.1 TRANSFERABILITY EXPERIMENTS ON GRAPH DATASETS

Subsampling protocol. The sampling procedure for induced graphs is the following: we random sample nodes uniformly. For any two points in the sampled set, if there exists an edge in the original graph, the induced subgraph includes that edge. Since the sampling is uniform, the label proportion is maintained in expectation. With respect to feature preprocessing, we use the features of each graph dataset as they are, with the exception of batch normalization at the input.

Datasets. SNAP-Patents is a network for patents granted between 1963 and 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.

Model implementation details. For both dense and sparse GTs, we implement multiheaded, scaled dot-product self-attention (Vaswani et al., 2017) with feedforward layers. RPEARL's GNNs are implemented as in Equation (2) using TAGConv filters (Du et al., 2017).

Table 1: Datasets used for transferability experiments. Edges are directed edges excluding self-loops; for OGBN-MAG we use the paper–paper citation subgraph. Max train/test nodes are the sizes of the full training and test graphs ($\alpha = 1$), with 10%/45% of nodes held out for validation/test (20%/40% for Squirrel).

Dataset	Nodes	Edges	Max Train/Test Nodes	Classes	Graphs	Feature Dim
Squirrel	5,201	216,933	2,081	5	1	2,089
ArXiv-year	169,343	1,166,243	76,205	5	1	128
OGBN-MAG	736,389	5,416,271	331,376	349	1	128
SNAP-Patents	2,923,922	13,975,788	1,315,765	5	1	269
Reddit-Binary	avg. 429.63	avg. 497.75	/	2	2000	1*

*Reddit-Binary has no node features; a constant feature is used.

Dataset preparation. Each dataset is split into train/val/test fractions of 45%-10%-45% 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 6, where the largest testing graph $\alpha = 1.0$ corresponds to using the full 45% training fraction and 45% of the testing fraction.

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. For each dataset, we tuned hyperparameters by training each model on the full training graph. Then, we held the selected hyperparameters fixed across all smaller training graph sizes. In general, we tuned learning rate over the interval $[1 \times 10^{-5}, 1 \times 10^{-2}]$, dropout and attention dropout over $[0, 0.5]$, the feedforward hidden dimension in $\{128, 256, 512\}$, the number of transformer heads in $\{4, 8\}$, the number of GNN/Transformer layers in $\{3, 4, 5, 6, 7\}$, the number of GNN/Sparse GT hops in $\{2, 3, 4\}$, and the Expander degree for Exphormer in $\{3, 4\}$. The full set of hyperparameters used for each model and dataset is available in Table 2.

Computational resources The experiments reported in this paper were executed on NVIDIA DGX B200 instances, using a single GPU per experiment (180GB of HBM3e Memory per process). Preliminary hyperparameter tuning was performed on a machine with two NVIDIA GeForce RTX 3090 GPUs, with 24GB of GPU memory each.

Table 2: Hyperparameters for each architecture and dataset. PE: positional encoding; for RPEARL, "Random" feeds random node signals and "Data" the node features into the PE network.

Model/Hyperparameters	Squirrel	ArXiv-year	OGBN-MAG	SNAP-Patents	Reddit-Binary
GNN					
Learning rate	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-3}
Dropout	0.5	0.5	0.25	0.25	0
Hidden channels	256	256	128	256	512
Number of hops	3	3	3	3	3
Number of layers	2	3	7	7	2
Max epochs	300	300	700	800	350
Dense GT + Eigenvector PE					
Learning rate	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Dropout	0.25	0.05	0.25	0.05	0.1
Attention dropout	0.5	0.15	0.15	0.15	0.1
Heads	8	4	4	4	8
Dim model	64	256	128	256	128
Dim feedforward	128	128	384	128	512
Number of layers	1	3	3	3	3
Eigenvectors (k)	16	8	8	8	8
Max epochs	300	300	700	300	350
Dense GT + RPEARL					
Learning rate	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Dropout	0.25	0.05	0.05	0.05	0.1
Attention dropout	0.5	0.15	0.15	0.15	0.1
Heads	8	4	4	4	8
Dim model	64	256	128	256	128
Dim feedforward	128	128	384	128	512
Number of layers	1	3	3	3	3
PE input	Random	Random	Data	Data	Random
PE hidden channels	128	128	128	128	512
PE number of layers	3	8	8	8	4
PE random samples	100	30	–	–	30
Max epochs	300	300	700	300	350
Sparse GT + Eigenvector PE					
Learning rate	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-5}
Dropout	0.01	0.01	0.5	0.01	0.01
Attention dropout	0.5	0.01	0.05	0.01	0.01
Heads	8	8	4	8	8
Dim model	64	256	128	256	128
Number of hops	1	2	2	2	2
Number of layers	1	3	3	8	3
Eigenvectors (k)	4	8	8	8	8
Max epochs	300	300	4000	1500	350
Sparse GT + RPEARL					
Learning rate	1×10^{-3}	5×10^{-4}	5×10^{-3}	5×10^{-4}	1×10^{-5}
Dropout	0.01	0.01	0.3	0.01	0.01
Attention dropout	0.5	0.01	0.25	0.01	0.01
Heads	8	8	4	8	8
Dim model	64	256	128	256	128
Number of hops	1	2	2	2	2
Number of layers	1	3	4	8	3
PE hidden channels	128	128	128	128	128
PE number of layers	3	8	1	8	8
PE random samples	30	30	5	30	30
Max epochs	300	300	700	1500	350

G Additional Experimental Results

G.1 Additional datasets and architectures

In this section, we provide additional results, including transferability analysis for other reference architectures, as well as the full heatmaps for every model-dataset combination.

Additional models. In addition to Dense GT and Sparse GT, we consider MLP and GNN (Du et al., 2017) baselines, GraphGPS (Rampášek et al., 2022), a GNN-transformer hybrid with Laplacian eigenvector PEs, and Exphormer (Shirzad et al., 2023), a sparse variant of Graph Transformer 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. Crucially, Exphormer has no positional encodings beyond its masking.

Transferability and comparison between models. Figure 4 contains transferability plots for all architectures. We observe transferability patterns for Dense GT, Sparse GT and GNN in most datasets. Exphormer shows transferable behavior in OGBN-MAG and SNAP-Patents. In all datasets, GT is either comparable (OGBN-MAG, Reddit-Binary) or significantly superior (SNAP-Patents, ArXiv-year) to GNN. This performance gap is consistent with previous work's observations of the success of global attention in heterophilic datasets (Dwivedi & Bresson, 2020). 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. In OGBN-MAG and Reddit-Binary, GNN and GTs show comparable performance and transferability patterns. In the cases of SNAP-Patents and OGBN-MAG, Exphormer presents a monotonically increasing performance as training graph size increases.

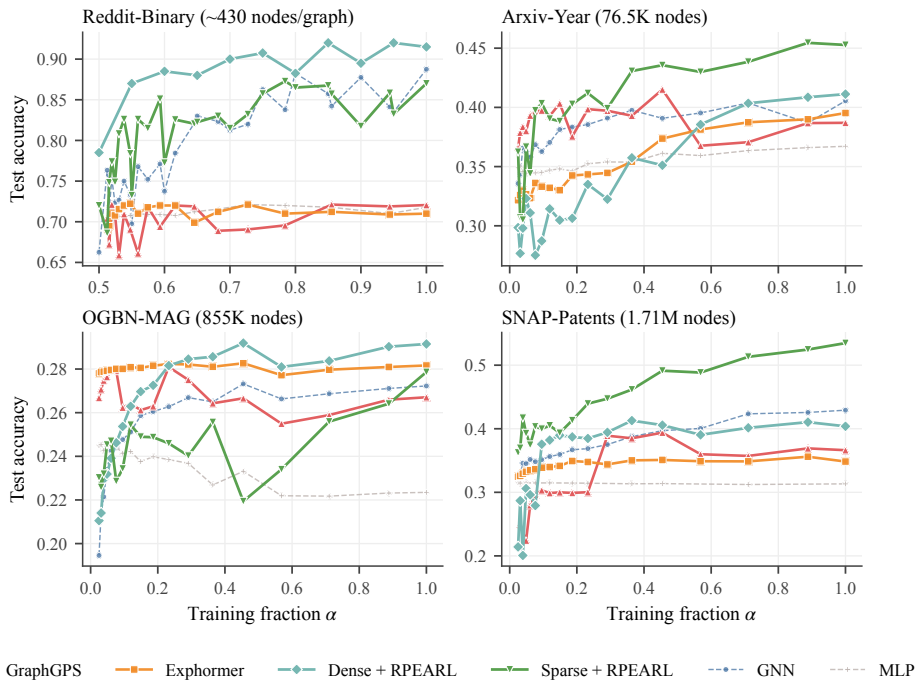


Figure 4: Transferability plots for all datasets and architectures. For each dataset, the x axis represents the train graph sizes as a proportion of the largest graph (α), and the y axis is the test accuracy at the full-sized graph. The titles show dataset name and largest graph size.

Transference to smaller test fractions. Transferability also implies that model performance transfers to graph sizes smaller than the training graph size. Figure 6 illustrates this effect with a

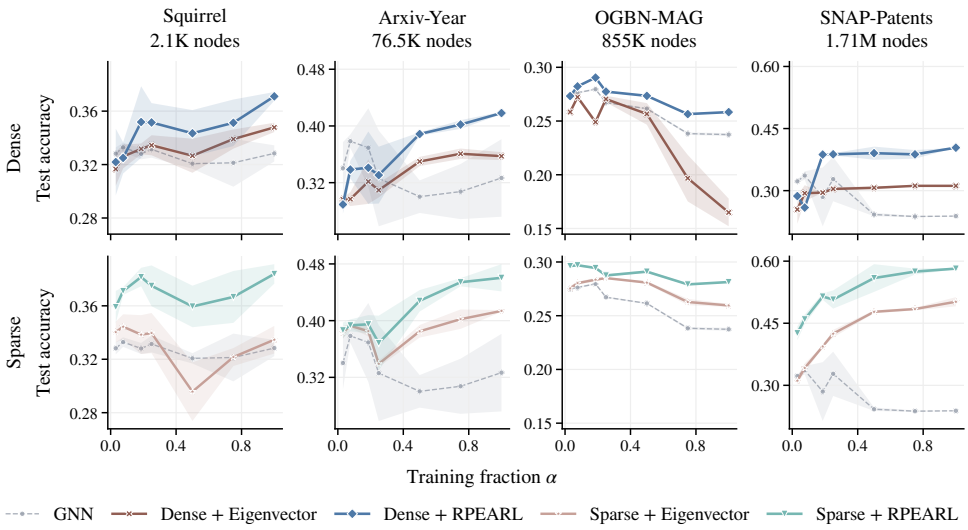


Figure 5: ERROR BARS

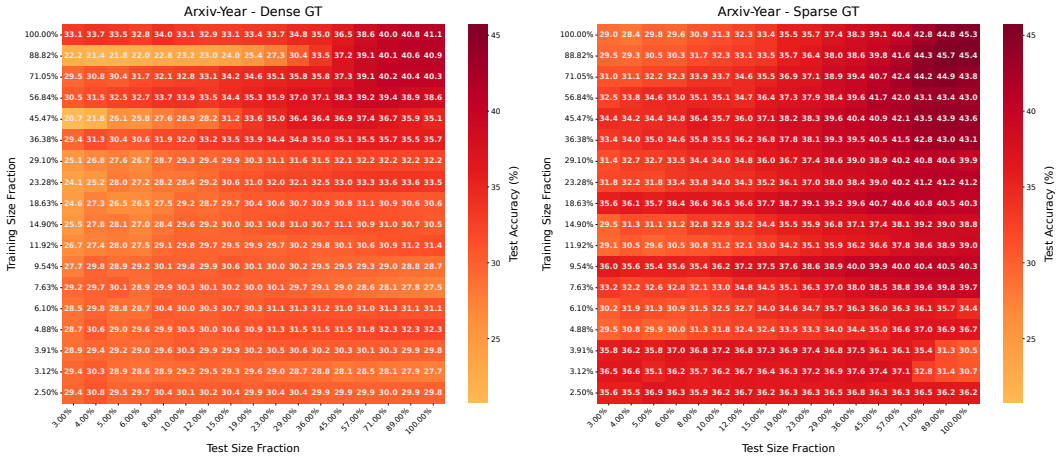


Figure 6: Transferability heatmaps on ArXiv-year. The x axis corresponds to train graph sizes as a proportion of the largest graph, and the y axis to test graph size fractions. The color corresponds to the test accuracy at each setting.

heatmap of test accuracy at various combinations of training (row) and testing (column) graph sizes. Both sparse and dense GT exhibit competitive accuracy at various settings.

G.2 Ablation analysis and effect of masking

RPEARL + Masking achieves the best transference on SNAP-Patents. We perform an ablation analysis of Sparse GT's structural components on the SNAP-Patents dataset to gain further insight into the effect of each mechanism on its transferability. For this, we train each architecture on a 30% downsample (513K nodes) of the largest training set, and evaluate on a 1.71M node test set. In addition to ablating RPEARL and Mask, we also consider random edges (RE) using random expander graphs of degree 3 (Shirzad et al., 2023). On Table 3, we can observe that RPEARL encodings enhance the baseline GT performance by 10.53%. Adding the sparse encoding mask improves performance by 26.68%. Combining both encodings yields a performance gain of 58.63%, reaching the best performance across all settings. Finally, random edges do not seem to improve performance.

Table 3: Test accuracy of GT trained with $\alpha = 0.3$, with different components in SNAP-Patents (1.71M nodes).

Architecture	Accuracy	% vs. GT
GT (no PE)	31.33	–
GT + RPEARL	34.63	+10.53%
GT + Mask	39.69	+26.68%
GT + Mask + RPEARL	**49.70**	**+58.63%**
GT + Mask + RE	31.01	-1.02%
GT + Mask + RPEARL + RE	46.55	+48.58%

Our findings coincide with observations from previous works that this attention mask may be a useful inductive bias (Lin et al., 2024).

k-hop parameter ablation. The k-hop neighborhood mask of Sparse GT trades off computational complexity and global information, resulting in up to 100x training speedups relative to Dense GT, while retaining comparable performance (see Table 4). Beyond this tradeoff, we also observed improved test performance in the previous ablation. In our experiments, we find all values of k to be similarly transferable, with the best performance in $k \in \{3, 4\}$. In Figure 7, we observe little sensitivity to the value of k. In this particular case, the optimal value is at $k = 4$.

G.3 Runtime analysis

Our transferability results imply that it is possible to train with small graphs and attain competitive test performance on larger graphs, resulting in shorter training times. In addition to this, Sparse GT reduces the computational complexity from quadratic in the number of nodes to quadratic in the k-hop neighborhood cardinality. Table 4 shows the training times of Dense and Sparse GT for multiple fractions. We observe that Sparse GT is one or two orders of magnitude faster than Dense GT in all settings. It is worth highlighting that Sparse GT is both faster and achieves superior performance than Dense GT in ArXiv-year, and SNAP-Patents.

G.4 Alternative Positional Encodings

Our experiments from Section 6 focused on analyzing the transferability of a variety of combinations of GT backbone and PE architecture. In this section we provide additional results on the transferability of multiple PE variants for a fixed GT backbone. In Figure 8a, we examine the transferability with different GNN architectures, all using RPEARL-style random features as described in Appendix D.6.

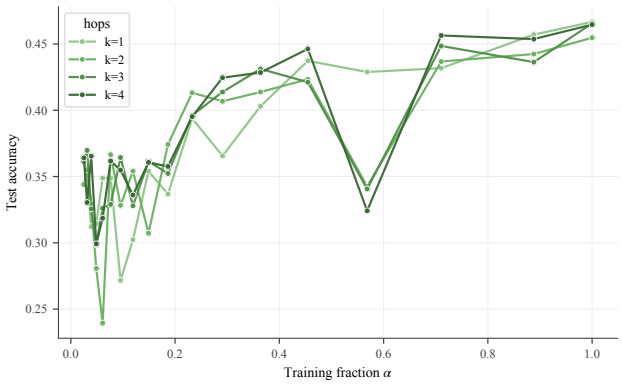
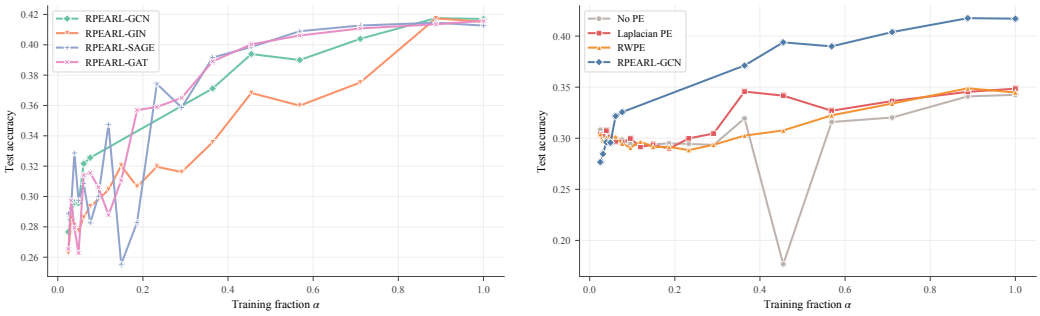


Figure 7: Transferability plot for Sparse GT on ArXiv-year for various values of the neighborhood parameter k.

Table 4: Runtimes in minutes for three subsample sizes of Sparse and Dense GT

Dataset	Model	6.10%	56.84%	100.00%
ArXiv-year	Dense GT	51.6	10.0	19.8
ArXiv-year	Sparse GT	8.8	1.4	2.0
OGBN-MAG	Dense GT	41.8	103.3	366.6
OGBN-MAG	Sparse GT	22.4	7.4	15.0
SNAP-Patents	Dense GT	223.3	945.7	2763.6
SNAP-Patents	Sparse GT	28.3	13.5	23.2



(a) Transferability of RPEARL with different GNNs for the positional encoding backbones. Dataset: ArXiv-year.

(b) Transferability of different positional encodings with the same GT backbone (dense). Dataset: ArXiv-year.

We observe similar transferability in all cases, with the exception of GIN, where we observe a slight degradation across training sizes.

We also compare the effect of different families of PE in Figure 8b. We observe similar performance across training fractions for RWPE and Laplacian-based PEs. Relative to these, RPEARL PEs exhibit improved performance across all training fractions.

G.5 Terrain graphs

In Section 6.2 we presented an application of our theoretical results to the problem of distance estimation over terrain graphs. Terrain graphs are point cloud approximations of a physical manifold. Some time-critical applications require low latency Shortest Path Distance (SPD) estimations between two points on a terrain, which is challenging in high-resolution point clouds with millions of nodes. It is also required that SPD estimations are computed accurately over various resolutions. This has been tackled in deep learning as a *metric learning* problem, where the task is to learn a latent space using a GNN in which the embedding distance between two points approximates the true SPD between them.

Sparse GT offers an efficient, scalable, and expressive architecture that can be applied to the metric learning problem in terrain graphs. Furthermore, the transferability guarantees of Corollary 2 imply that we need not train with the highest possible resolution of terrain graph in order to achieve competitive performance. Here we provide implementation details and an additional result on a second terrain graph dataset.

Metric learning. The setup of (Chen et al., 2025) is to use imitation learning to learn a latent space that resembles true shortest-path distance between points. For two points x_i, x_j in the terrain graph, we train to minimize $\|L_1(\phi(x_i) - \phi(x_j)) - SPD(x_i, x_j)\|_2$, where ϕ is an embedding function, in our case either GNN or SGT. On (Chen et al., 2025), a subsequent stage consists of freezing the embedding function's parameters and finetuning an additional MLP on a larger set of sample points. In our experiments, we omit this second stage. During training, we bias the point pair sampling to increase the frequency of difficult paths (points with higher altitude variance and shortest distance).

Datasets. The terrain graphs we use are Digital Elevation Model (DEM) datasets. The results of Section 6.2 used a DEM of the Troms region of Norway (Kartverket (Norwegian Mapping Authority),

2025). The Norway dataset is available in a 2000×2000 (4M nodes) resolution. For our experiments we downsample so that the largest grid graph is 250×250.

Graph and train/test data constructions. To test transferability, we generate graphs of evenly spaced points based on the high resolution graph. The downsampling parameter r controls the space between generated points, i.e. $r = 1$ is the full resolution, $r = 2$ takes every second point (resulting in a 1000×1000 graph), etc. The edges of the graph are taken to be the 8 nearest neighbors. For training datasets, we subsample 500 sources and 100 targets per source, for a total of $50,000$ pairs. For testing, a total of $10,000$ source points are selected based on maximum height, and the SPD is computed between these points and every other point in the graph.