

# Convergence of Graph Neural Networks on Relatively Sparse Graphs

Zhiyang Wang   Luana Ruiz   Alejandro Ribeiro

**Abstract**—In this paper we study the connection between graph filters, graph neural networks (GNNs) and manifold filters, manifold neural networks (MNNs). Specifically, we consider the case when we have access to a set of uniformly sampled points from the manifold based on which we construct a relatively sparse graph to approximate the manifold, which is a suitable model for many real world applications. We prove a non-asymptotic approximation error bound or convergence rate for the graph filters and the GNNs on the relatively sparse graphs to the filters and neural networks on the manifold that the graphs are sampled from. An interesting trade-off between the convergence and the discriminability of the graph filters can be observed from the non-asymptotic error bound which indicates that graph filters cannot give good convergence and discriminability at the same time. While the nonlinearity function in GNNs can alleviate this trade-off and allows the GNNs to both converge to the MNNs and discriminate well. Equipped with this non-asymptotic error bound, we further interpret the transferability property of GNNs when the graphs are sampled from a common manifold. We verify our conclusions with a point-cloud classification problem.

**Index Terms**—Graph neural networks, manifold convolution, manifold neural networks, relatively sparse graphs, convergence rate, transferability analysis

## I. INTRODUCTION

Modern signal processing now has shown increasing interest in data supported on geometric structures in non-Euclidean domains. This is motivated by a large number of applications, including but not limited to robot flocking [1], [2], molecular representations [3], [4], 3D shape analysis [5], [6] and wireless resource allocation [7], [8]. Graphs and manifolds are most commonly used to model the data structures in non-Euclidean domains [9]. Convolutional filters and convolutional neural networks, as the standard invariant and stable information processing tools which also allow feature sharings [10], have been established soundly on graphs as well as manifolds. Graph convolutional filters [11], [12], graph neural networks (GNNs) [13]–[15] together with manifold convolutional filters [16], [17] and manifold neural networks (MNNs) [18], [19] are therefore the prominent choices for non-Euclidean information processing in discrete and continuous domains respectively.

The relationship between graphs and manifolds have been studied which claims that graphs can be seen as discretizations of the manifolds if the graphs have a well-defined limit [9], [20]. Manifolds, as continuous latent spaces, are often accessed by a set of discrete sampled points over the manifolds [6], [16],

[19]. Based on these sampled points, a graph model can be built involving the local and global geometric information to give an approximation to the underlying manifold [19], [21]. Recent works have proved that convolutional filters and neural networks on the graphs sampled from the manifold converge to the manifold filters and MNNs both experimentally [9], [19] and theoretically with an asymptotic convergence result [19].

In this paper we focus on proposing a non-asymptotic convergence result to measure the approximation error between graph filters and GNNs on sampled graphs from the underlying manifold to the manifold filters and MNNs. Especially, the sampled graphs are constructed as random geometric graphs [22] where only nodes that are close enough are connected with edges, which makes the graph model more realistic. Plus, finite-sample error bounds can help determine the design and the number of sampled points needed of the sampled graphs while satisfying a given approximation error tolerance. The non-asymptotic approximation error bounds can also reveal some phenomena in the convergence regime which cannot be observed from the asymptotic counterparts, e.g. convergence rates, convergence trade-offs and the transferability property.

To close the gap between GNNs and MNNs, we start with the introductions of graph signal processing and GNNs along with manifold signal processing and MNNs. We employ the manifold convolutional filter defined as the integration of the Laplace-Beltrami (LB) operator exponentials which has been proved consistent with the graph convolutional filter and the standard time convolutional filter [17]. We then describe how to build a relatively sparse graph based on the sampled points from the underlying manifold and propose that the constructed graph Laplacian can approximate the LB operator of the manifold in both the operator and spectral aspects (Proposition 1 and 2). We present the approximation error bound (or the convergence rate) of the graph filters on the constructed relatively sparse graph to the manifold filters in Theorem 1 under certain assumptions, from where we can observe a trade-off between the convergence rate and the discriminability of the graph filter. The approximation error bound is derived similarly for GNNs by cascading graph filters with nonlinearities, which help alleviate the trade-off and enable GNNs to both converge and discriminate well. The transferability property for GNNs on different sampled graphs from the same underlying manifold can also be analyzed based on the non-asymptotic convergence result, which we also verify empirically with numerical experiments.

GNNs have been comprehensively discussed in many works [12]–[14]. The convergence and transferability of GNNs have been analyzed with the graphon model as the limit of a

Supported by NSF CCF 1717120, Theorinet Simons. Zhiyang and Alejandro are with Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, USA. Luana is with Computer Science & Artificial Intelligence Laboratory, MIT, Massachusetts, USA.

sequence of graphs [23]–[26]. Our paper instead sees the limit of large graphs as a manifold, which is more intuitive and general compared with the graphon model. In [19], a two-way connection between GNNs and MNNs are established, which states that MNNs can recover GNNs by discretizing in space and time domains, and GNNs converge to MNNs asymptotically as the size of the graph increases. However, there lacks an explicit convergence rate. In [27], a convergence rate is stated for the GNNs when approximating MNNs, but the conclusion is restrained to input signals within a limited bandwidth. Plus the graphs constructed in [19] and [27] still need to be dense graphs. In our work, we lift the bandlimited assumption of the input signals by importing a frequency dependent filter and we put our focus on building a relatively sparse graph which is a more practical model.

The rest of the paper is organized as follows. We start with preliminary concepts of graph signal processing and manifold signal processing in Section II. We construct the relatively sparse graphs by sampled points from the manifold in Section III. In Section IV, we present the approximation error bounds of graph filters and GNNs on the relatively sparse graphs to the manifold filters and MNNs respectively. Our proposed results are verified in a model classification problem in Section V. The conclusions are presented in Section VI.

## II. PRELIMINARIES

We begin by revising the basic concepts of graph signal processing, graph neural networks and manifold signal processing, manifold neural networks.

### A. Graph Signal Processing and Graph Neural Networks

Let  $\mathbf{G}$  be an undirected graph with  $n$  nodes. Graph signals  $\mathbf{x} \in \mathbb{R}^n$  are data supported on the nodes of the graph. The edge weights are given by a weight function with which we can get the matrix representation or Graph Shift Operator (GSO)  $\mathbf{S}$  of graph  $\mathbf{G}$  (i.e. adjacency matrix, Laplacian matrix) [11], [28]. The graph convolution is defined as the summation of the iterative graph diffusion process [11], [12]. It can be written explicitly as the polynomial of the GSO as

$$\mathbf{y} = \mathbf{h}_{\mathbf{G}}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}. \quad (1)$$

Considering the case of an undirected graph, the GSO is symmetric and admits an eigenvector decomposition as  $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$ , with orthogonal eigenvector matrix  $\mathbf{V} \in \mathbb{R}^{n \times n}$  and diagonal eigenvalue matrix  $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ . Suppose the diagonal eigenvalue entries are ordered as  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ . By projecting the graph filter output on the eigenvector matrix, we can write the spectral representation of the graph convolutional filter, which is explicitly denoted as

$$\mathbf{V}^H \mathbf{y} = \mathbf{V}^H \mathbf{h}_{\mathbf{G}}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{\Lambda}^k \mathbf{V}^H \mathbf{x} = \hat{h}(\mathbf{\Lambda}) \mathbf{V}^H \mathbf{x}. \quad (2)$$

The frequency response of the graph convolution can be defined as  $\hat{h}(\lambda) = \sum_{k=0}^{K-1} h_k \lambda^k$ , which relates the frequency

components of input and output graph signals point-wisely, with a full dependence on the filter coefficients  $\{h_k\}_{k=1}^{K-1}$  and the eigenvalues of  $\mathbf{S}$ .

A graph neural network (GNN) is composed of layers of graph filter banks followed by a nonlinearity function  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ . In the  $l$ -th layer of a GNN, the filters produce  $F_l$  features  $\mathbf{x}_l^p$ , which can be written explicitly as

$$\mathbf{x}_l^p = \sigma \left( \sum_{q=1}^{F_{l-1}} \mathbf{h}_{\mathbf{G}}^{lpq}(\mathbf{S}) \mathbf{x}_{l-1}^q \right), \quad (3)$$

for every  $1 \leq q \leq F_{l-1}$ ,  $1 \leq p \leq F_l$ . The number of features in each layer  $l = 1, 2, \dots, L$  is denoted as  $F_l$ . The graph filter  $\mathbf{h}_{\mathbf{G}}^{lpq}$  is as (1) defined which maps the  $q$ -th feature in the  $l-1$ -th layer to the  $p$ -th feature in the  $l$ -th layer. To simplify the presentation, we denote the GNN composed of  $L$  layers with (3) as a map  $\Phi_{\mathbf{G}}(\mathbf{H}, \mathbf{S}, \mathbf{x})$ , where  $\mathbf{H}$  includes all the graph filter coefficients across all the layers.

### B. Manifold Signal Processing and Manifold Neural Networks

We consider a  $d$ -dimensional compact, smooth and differentiable embedded manifold  $\mathcal{M} \in \mathbb{R}^N$  with measure  $\mu$ . Manifold signals are scalar functions supported over  $\mathcal{M}$  as  $f : \mathcal{M} \rightarrow \mathbb{R}$  [16], [17]. The local Euclidean space around each point  $x \in \mathcal{M}$  is called tangent space and the disjoint union of all these tangent spaces over  $\mathcal{M}$  is defined as *tangent bundle*  $T\mathcal{M}$ . The Laplace-Beltrami (LB) operator  $\mathcal{L} : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{M})$  is defined as the intrinsic divergence of the intrinsic gradient of the scalar functions over  $\mathcal{M}$ , which can be explicitly written as

$$\mathcal{L}f = -\text{div} \circ \nabla f. \quad (4)$$

Like graph Laplacians [29], this LB operator measures the difference between the function value at some point and the average value around this point [9].

Manifold convolution can be defined as the integration of a heat diffusion process over the manifold [17]. This way of defining the manifold convolution has been proved consistent with the graph convolution and standard time convolution [19] which enables us to connect graphs and manifolds. With  $\tilde{h} : \mathbb{R}^+ \rightarrow \mathbb{R}$  denoted the filter impulse function, the manifold convolutional filter can be written as

$$g(x) = \int_0^\infty \tilde{h}(t) e^{-t\mathcal{L}} f(x) dt = \mathbf{h}(\mathcal{L}) f(x). \quad (5)$$

Considering that the LB operator is self-adjoint and positive-semidefinite, it has real and positive eigenvalues. Plus the compactness of  $\mathcal{M}$ , the LB operator possesses discrete spectrum, which can be denoted as pairs of eigenvalues and eigenfunctions  $\{\lambda_i, \phi_i\}_{i=1}^\infty$  with the order  $\lambda_1 \leq \lambda_2 \leq \lambda_3 \dots$ . By projecting the output of the manifold convolution onto the eigenfunction  $\phi_i$ , we can get the spectral representation as

$$[\hat{g}]_i = \int_0^\infty \tilde{h}(t) e^{-t\lambda_i} dt [\hat{f}]_i = \hat{h}(\lambda_i) [\hat{f}]_i, \quad (6)$$

where  $[\hat{f}]_i = \int_{\mathcal{M}} f(x) \phi_i(x) d\mu(x)$  is the frequency component of function  $f$ . Therefore, we can see the filter frequency response

is point-wise on each frequency component which can be represented as  $\hat{h}(\lambda) = \int_0^\infty \hat{h}(t)e^{-t\lambda}dt$ . With the function can be represented on the eigenfunction basis as  $g = \sum_{i=1}^\infty [\hat{g}]_i \phi_i$ , the spectral representation of the manifold filter is

$$g = \sum_{i=1}^\infty [\hat{g}]_i \phi_i = \sum_{i=1}^\infty \hat{h}(\lambda_i) [\hat{f}]_i \phi_i. \quad (7)$$

Similar to graph convolutions, the manifold filter frequency response also fully depends on the filter impulse function and the eigenvalues of the LB operator.

A manifold neural network (MNN) can be likewise defined as a the cascading layers of manifold filter banks point-wise non-linearities  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ . The output of the  $l$ -th layer can be explicitly written as

$$f_l^p(x) = \sigma \left( \sum_{q=1}^{F_{l-1}} \mathbf{h}_l^{pq}(\mathcal{L}) f_{l-1}^q(x) \right), \quad (8)$$

where  $f_l^p$ ,  $1 \leq p \leq F_l$  is the  $p$ -th feature in the  $l$ -th layer. The manifold filter  $\mathbf{h}_l^{pq}$  in the  $l$ -th layer maps  $F_{l-1}$  input features to  $F_l$  output features. To be more concise, we denote the MNN as  $\Phi(\mathbf{H}, \mathcal{L}, f)$  where the function set  $\mathbf{H}$  includes all the impulse response functions of the manifold filters  $\mathbf{h}_l^{pq}$  of all layers.

### III. SPARSE GRAPH CONSTRUCTION FROM MANIFOLDS

We can access a continuous manifold  $\mathcal{M}$  by processing a set of discrete uniformly sampled points from  $\mathcal{M}$ . By connecting these sampled points as a graph, we can get an approximation of the underlying manifold. With the sampled points seen as the nodes of the graph, the weight values of the edges connecting the nodes are defined based on the Euclidean distance between the nodes. Let  $X$  be the set of  $n$  i.i.d. sampled points  $\{x_1, x_2, \dots, x_n\}$ . Explicitly, we construct an undirected graph  $\mathbf{G}_n$  with points  $X$  as nodes and the weight value  $w_{ij}$  between  $x_i$  and  $x_j$  defined as

$$w_{ij} = \frac{1}{n} \frac{d+2}{\epsilon^{d/2+1} \alpha_d} \mathbb{I}_{[0,1]} \left( \frac{\|x_i - x_j\|^2}{\epsilon} \right), \quad (9)$$

where  $\|x_i - x_j\|$  denotes the Euclidean distance between  $x_i$  and  $x_j$  while  $\alpha_d$  is the unit ball volume in  $\mathbb{R}^d$ . The indicator function shows that two nodes can only be connected if they are smaller than  $\sqrt{\epsilon}$  away from each other. According to the random geometric graph theory [22], the order of  $\epsilon$  affects the average degree of the nodes. That is, if  $\epsilon$  is in the order of  $O((\log(n)/n)^{2/d})$ , the average node degree is  $O(\log(n))$ , which falls in a relatively sparse regime. The graph Laplacian  $\mathbf{L}_n^\epsilon$  is defined as  $\mathbf{L}_n^\epsilon = \text{diag}(\mathbf{A}_n \mathbf{1}) - \mathbf{A}_n$  [30] with  $\mathbf{A}_n, [\mathbf{A}_n]_{ij} = w_{ij}$  standing for the adjacency matrix.

The relationship of graph Laplacian and LB operator can be quantified in both the operator and spectral aspects. A non-asymptotic difference bound can be obtained between the graph Laplacian and the LB operator when they both operate on the eigenfunctions of the LB operator.

**Proposition 1** [31, Theorem 3.3] Suppose that  $\mathcal{M} \in \mathbb{R}^N$  is equipped with LB operator  $\mathcal{L}$  with spectrum  $\{\lambda_i, \phi_i\}_{i=1}^\infty$  and the

graph  $\mathbf{G}_n$  sampled from  $\mathcal{M}$  is equipped with graph Laplacian  $\mathbf{L}_n^\epsilon$  with edge weights set as (9) with  $\epsilon = \epsilon(n) > (\log(n)/n)^{2/d}$ . Then with probability at least  $1 - \delta$ , it holds that

$$|\mathbf{L}_n^\epsilon \phi_i(x) - \mathcal{L} \phi_i(x)| \leq \left( C_1 \sqrt{\frac{\ln(2n/\delta)}{cn\epsilon^{d+2}}} + C_2 \sqrt{\epsilon} \right) \lambda_i^{\frac{d+2}{4}}. \quad (10)$$

The constants  $C_1, C_2$  depend on the volume of the manifold.

This point-wise upper bound indicates that when applied to the eigenfunction  $\phi_i$ , the difference is related to the number of sampled points  $n$  as well as the corresponding eigenvalue  $\lambda_i$ . This is due to the fact that eigenfunctions with higher eigenvalues change faster and are harder to measure the differences in high frequency domain [32].

With this operator difference bound, the difference bounds of the spectrum of graph Laplacian  $\mathbf{L}_n^\epsilon$  and the LB operator  $\mathcal{L}$  can also be derived based on Davis-Khan theorem [33].

**Proposition 2** [31, Theorem 2.4, Theorem 2.6] Suppose that  $\mathcal{M} \in \mathbb{R}^N$  is equipped with LB operator  $\mathcal{L}$  with spectrum  $\{\lambda_i, \phi_i\}_{i=1}^\infty$  and the graph  $\mathbf{G}_n$  sampled from  $\mathcal{M}$  is equipped with graph Laplacian  $\mathbf{L}_n^\epsilon$  with edge weights set as (9) with  $\epsilon = \epsilon(n) > (\log(n)/n)^{2/d}$ , with the spectrum given by  $\{\lambda_{i,n}^\epsilon, \phi_{i,n}^\epsilon\}_{i=1}^n$ . Fix some  $K \in \mathbb{N}^+$ , then with probability at least  $1 - 2n \exp(-cn\epsilon^{d/2+2})$ , we have

$$|\lambda_i - \lambda_{i,n}^\epsilon| \leq C_{K,1} \sqrt{\epsilon}, \quad \|a_i \phi_{i,n}^\epsilon - \phi_i\| \leq C_{K,2} \sqrt{\epsilon}/\theta, \quad (11)$$

with  $a_i \in \{-1, 1\}$  for all  $i < K$  and  $\theta$  the eigengap of  $\mathcal{L}$ , i.e.  $\theta = \min_{1 \leq i \leq K} \{\lambda_i - \lambda_{i-1}, \lambda_{i+1} - \lambda_i\}$ . The constants  $C_{K,1}, C_{K,2}$  depend on  $\lambda_K, d$  and the volume of  $\mathcal{M}$ .

We can observe from Proposition 2 that the eigenvalue and eigenfunction difference bounds can only be given within a limited spectrum, i.e.  $\lambda_i < \lambda_K$ . This is due to the fact that eigenfunctions oscillate faster in high frequency domain and are hard to approximate. This indicates that we need to pay special attention to the high frequency domain when designing the filters as the graph filters need to converge as well as discriminate different frequency components.

We note that in the case of Proposition 1 and 2,  $\epsilon$  is in the order of  $(\log(n)/n)^{2/d}$ , which leads to the average node degree scales with the order  $\log(n)$ . This ensures that our constructed graphs are relatively sparse when  $d > 2$ .

### IV. CONVERGENCE OF GNNs ON SPARSE GRAPHS

As we have the definition of manifold convolutional filter parametric by the LB operator as (5) shows, we can transfer this filter structure to the discrete graph Laplacian and approximate the manifold filter with this newly defined graph filter. This approximation can be quantified with a non-asymptotic error bound, which can also be seen as the convergence rate of the graph filter to the manifold filter. GNN, as a cascading structure of graph filters and point-wise nonlinearities, can approximate MNN with the convergence and approximation inherited from the graph filters.

### A. Graph Convolution on Sampled Manifolds

By fixing the filter impulse function  $\tilde{h}(t)$  in equation (5) and replacing the LB operator  $\mathcal{L}$  with the discrete graph Laplacian  $\mathbf{L}_n^\epsilon$ , we can write a graph filtering process in a continuous time domain (instead of the discrete time domain defined in (1)), which explicitly is

$$\mathbf{g} = \int_0^\infty \tilde{h}(t) e^{-t\mathbf{L}_n^\epsilon} \mathbf{f} dt := \mathbf{h}(\mathbf{L}_n^\epsilon) \mathbf{f}, \quad \mathbf{g}, \mathbf{f} \in \mathbb{R}^n. \quad (12)$$

We can see this as an integration of the graph shift operations with the GSO represented as  $e^{-\mathbf{L}_n^\epsilon}$ . The discrete graph signal  $\mathbf{f}$  can be obtained by a uniform sampling operator  $\mathbf{P}_n$  operated on manifold signal  $f$ , i.e.  $\mathbf{f} = \mathbf{P}_n f$  with  $\mathbf{f}(x_i) = f(x_i)$ ,  $x_i \in X$ .

With the spectrum of  $\mathbf{L}_n^\epsilon$  denoted as  $\{\lambda_{i,n}^\epsilon, \phi_{i,n}^\epsilon\}_{i=1}^n$ , the above graph filter can be reformalized in the spectral domain by projecting on the eigenvectors as

$$\mathbf{g} = \sum_{i=1}^n \hat{h}(\lambda_{i,n}^\epsilon) \langle \mathbf{f}, \phi_{i,n}^\epsilon \rangle_{L^2(\mathbf{G}_n)} \phi_{i,n}^\epsilon. \quad (13)$$

Together with (7), the spectral representations of the graph filter and the manifold filter both depend fully on the spectrum of the graph Laplacian and the LB operator respectively. Therefore, the connection between graph filtering and manifold filtering can be revealed based on the spectral relationship that we have established in Proposition 2. We first look at the convergence of graph filters to the manifold filters.

### B. Graph Convolution Convergence

Considering that the convergence result in Proposition 2 is limited within a range of spectrum while the spectrum of the LB operator is infinite, we need to import a *frequency dependent filter* to tackle the intractable frequency components in the high frequency domain. Weyl's law [34] reveals that the eigenvalues of the LB operator tend to accumulate in high frequency domain, which is explicitly stated as the following lemma.

**Lemma 1** [35, Proposition 3] *Consider a  $d$ -dimensional manifold  $\mathcal{M} \subset \mathbb{R}^N$  and let  $\mathcal{L}$  be its LB operator with eigenvalues  $\{\lambda_k\}_{k=1}^\infty$ . Let  $C_1$  be an arbitrary constant and  $\alpha_d$  the volume of the  $d$ -dimensional unit ball. Let  $\text{Vol}(\mathcal{M})$  denote the volume of manifold  $\mathcal{M}$ . For any  $\alpha > 0$  and  $d > 2$ , there exists  $N_1$ ,*

$$N_1 = \lceil (\alpha d / C_1)^{d/(2-d)} (C_d \text{Vol}(\mathcal{M}))^{2/(2-d)} \rceil \quad (14)$$

*such that, for all  $k > N_1$ ,  $\lambda_{k+1} - \lambda_k \leq \alpha$ .*

Based on the Weyl's law, we can implement a spectrum partition strategy as Definition 1 shows. The frequency difference threshold filter –  $\alpha$ -FDT filter defined in Definition 2 can realize the  $\alpha$ -separated spectrum.

**Definition 1** [35, Definition 4] ( $\alpha$ -separated spectrum) *The  $\alpha$ -separated spectrum of the LB operator  $\mathcal{L}$  is defined as a partition  $\Lambda_1(\alpha) \cup \dots \cup \Lambda_N(\alpha)$  if it holds that  $|\lambda_i - \lambda_j| > \alpha$  for  $\lambda_i \in \Lambda_k(\alpha)$  and  $\lambda_j \in \Lambda_l(\alpha)$ ,  $k \neq l$ .*

**Definition 2** [35, Definition 5] ( $\alpha$ -FDT filter) *The  $\alpha$ -frequency difference threshold ( $\alpha$ -FDT) filter is defined as a filter  $\mathbf{h}(\mathcal{L})$  whose frequency response satisfies*

$$|\hat{h}(\lambda_i) - \hat{h}(\lambda_j)| \leq \gamma_k \text{ for all } \lambda_i, \lambda_j \in \Lambda_k(\alpha) \quad (15)$$

*with  $\gamma_k \leq \gamma$  for some  $\gamma > 0$  and  $k = 1, \dots, N$ .*

Furthermore, the filter frequency response function needs to be Lipschitz continuous as addressed in Definition 3.

**Definition 3** (Lipschitz filter) *A filter is  $A_h$ -Lipschitz if its frequency response is Lipschitz continuous with Lipschitz constant  $A_h$ , i.e.*

$$|\hat{h}(a) - \hat{h}(b)| \leq A_h |a - b| \text{ for all } a, b \in (0, \infty). \quad (16)$$

Equipped with these concepts and assumptions, the approximation error bound can be derived to measure the output difference between the graph filters and manifold filters, which further can attest the convergence of the graph filters to the manifold filters as presented in Theorem 1.

**Theorem 1** (Convergence of graph filters) *Suppose that  $\mathcal{M} \in \mathbb{R}^N$  is equipped with LB operator  $\mathcal{L}$  and the graph  $\mathbf{G}_n$  sampled from  $\mathcal{M}$  is equipped with graph Laplacian  $\mathbf{L}_n^\epsilon$  with edge weights set as (9) with  $\epsilon = \epsilon(n) > (\log(n)/n)^{2/d}$ . Let  $\mathbf{h}(\cdot)$  be the convolutional filter and assume the frequency response of filter  $\mathbf{h}$  is  $A_h$  Lipschitz continuous and  $\alpha$ -FDT with  $\alpha^2 \gg \epsilon$ ,  $\alpha > C_{\mathcal{M},d} K^{1-2/d}$  and  $\gamma = C'_{K,2} \epsilon / \alpha$ . Then with probability at least  $1 - 2n \exp(-C_n \epsilon^{d+4})$  it holds that*

$$\begin{aligned} \|\mathbf{h}(\mathbf{L}_n^\epsilon) \mathbf{P}_n f - \mathbf{P}_n \mathbf{h}(\mathcal{L}) f\|_{L^2(\mathbf{G}_n)} \\ \leq \left( \frac{N C'_{K,2}}{\alpha} + A_h C_{K,1} \right) \sqrt{\epsilon} + C_{gc} \sqrt{\frac{\log n}{n}} \end{aligned} \quad (17)$$

*where  $N$  is the partition size of  $\alpha$ -FDT filter and  $C_{gc}$  is related with  $d$  and the volume of  $\mathcal{M}$ .*

*Proof.* See Appendix A. ■

The approximation upper bound for the output difference between graph filters and manifold filters is in the order of  $O((\log(n)/n)^{1/d})$ . This attests the convergence of the graph filters on the relatively sparse graphs sampled from the underlying manifold with the convergence rate  $O((\log(n)/n)^{1/d})$ . Besides this, we can observe that the error bound in (17) grows with the manifold dimension  $d$ , which means that a higher dimension makes the graph filters harder to approximate the manifold filters. Moreover, the imported frequency dependent filters can remove the limitation on the spectrum by setting  $\alpha$  large enough to group all the eigenvalues larger than  $\lambda_K$  as one partition, with a smaller  $K$  leading to a larger  $\alpha$ . The  $\alpha$ -FDT filter gives similar frequency responses to eigenvalues in the same group, which can mitigate the divergence of high frequency components. However, there exists a trade-off for this benefit. If we fix the number of sampled points  $n$ , a larger  $\alpha$  (i.e. a smaller  $K$ ) leads to a less discriminative filter as more eigenvalues are supposed to be grouped and treated similarly, which makes the high frequency components within the same partition cannot

be discriminated. Meanwhile, a larger  $\alpha$  leads to a smaller number of partitions as the number of singletons decrease, which results in the decrease of the approximation error bound shown in (17). The Lipschitz continuity constant of the filters  $A_h$  affects approximation and discriminability a similar way. Smaller Lipschitz constants decrease the approximation error bound, but result in smoother filter functions that can give similar frequency responses to different eigenvalues even in different partitions. These indicate that graph filters cannot be discriminative and approximative to the manifold filters at the same time. That is to say, the graph filters cannot discriminate all frequency components well and converge to manifold filters fast at the same time. In the following we will show that this trade-off phenomenon can be alleviated by the nonlinearity functions in GNNs.

### C. Convergence of GNNs

As GNNs are cascading structures of graph filters and nonlinearities, they can inherit the convergence of graph filters shown in Theorem 1. We first impose an assumption on the continuity of nonlinearity function as the follows, which is satisfied by most common nonlinearities (e.g., the ReLU, the modulus and the sigmoid).

**Assumption 1** (Normalized Lipschitz nonlinearity functions) *The nonlinearity function  $\sigma$  is normalized Lipschitz continuous, i.e.,  $|\sigma(a) - \sigma(b)| \leq |a - b|$ , with  $\sigma(0) = 0$ .*

The approximation error bound or convergence rate of GNNs can be derived based on the result in Theorem 1, as presented in the following theorem.

**Theorem 2** *Suppose that  $\mathcal{M} \in \mathbb{R}^N$  is equipped with LB operator  $\mathcal{L}$  and the graph  $\mathbf{G}_n$  sampled from  $\mathcal{M}$  is equipped with graph Laplacian  $\mathbf{L}_n^\epsilon$  with edge weights set as (9) with  $\epsilon = \epsilon(n) > (\log(n)/n)^{2/d}$ . Let  $\Phi(\mathbf{H}, \mathcal{L}, \cdot)$  be an  $L$ -layer MNN on  $\mathcal{M}$  (8) with  $F_0 = F_L = 1$  input and output features and  $F_l = F$ ,  $l = 1, 2, \dots, L-1$  features per layer and  $\Phi(\mathbf{H}, \mathbf{L}_n, \cdot)$  be the MNN with the same architecture applied on geometric graph  $\mathbf{G}_n$ . The nonlinearity functions satisfy Assumption 1, it holds that*

$$\begin{aligned} & \|\Phi(\mathbf{H}, \mathbf{L}_n^\epsilon, \mathbf{P}_n f) - \mathbf{P}_n \Phi(\mathbf{H}, \mathcal{L}, f)\|_{L^2(\mathbf{G}_n)} \\ & \leq LF^{L-1} \left( \left( \frac{NC'_{K,2}}{\alpha} + A_h C_{K,1} \right) \sqrt{\epsilon} + C_{gc} \sqrt{\frac{\log n}{n}} \right) \end{aligned} \quad (18)$$

with high probability.

*Proof.* See [36]. ■

We now can come to the conclusion that the GNN on the constructed relatively sparse graphs can converge to the MNN in the order of  $O((\log(n)/n)^{1/d})$ . On one side, the approximation error bound scales with the size of the neural network architecture, which is due to the error propagation through the networks. Specifically, the bound grows linearly with the number of layers  $L$  and polynomially with the number of features  $F$  where the rate is determined by  $L$ . On the other

side, the approximation error bound or the convergence result inherits the trade-off possessed by the graph filters indicated in Theorem 1. However, the nonlinearity functions in GNN can mix the spectral components which means the high frequency components can be shifted to low frequency domain and can be later discriminated by the filters in the following layer. The effects of nonlinearity functions have been discussed in neural networks on graphs and manifold respectively in previous works [17], [37]. The effects brought by the nonlinearities make the GNN both approximative and discriminative, which lifts the trade-off inserted by the graph filters.

With the non-asymptotic approximation error bound derived, the transferability property for GNNs on the relatively sparse graphs with different size from a common underlying manifold can be immediately deduced based on the triangle inequality. This property can attest that a trained GNN on a small graph can be directly transferred to another different larger graph sampled from the same manifold as long as they are constructed with the same manner. Moreover, with the non-asymptotic result, we can calculate the minimum number of sampled points needed to satisfy the given approximation error tolerance. The transferability property is further verified with simulations in the following section.

## V. SIMULATIONS

We verify our proved convergence results on the ModelNet10 [38] classification problem. This dataset includes meshed CAD models from 10 different categories with 3,991 models for training and 908 models for testing. We construct graphs by sampling  $n$  points uniformly from the meshed models to approximate the underlying models as shown in Figure 1. The goal is to identify the chair models from other models.



Fig. 1: Graphs constructed with 300 sampled points

**Learning architectures and experiment settings.** We build the relatively sparse graphs with sampled points seen as nodes and the weights determined according to (9) with  $\epsilon = 0.001$  as the threshold. The graph Laplacian can be calculated accordingly. We compare the performances of three architectures, which includes 2-layer Graph Filters (GF), 2-layer graph neural networks (GNN) and 2-layer graph neural networks with Lipschitz continuous filters (Lipschitz GNN). Each architecture contains  $F_0 = 3$  input features which are each point's 3-d coordinates,  $F_1 = 64$  and  $F_2 = 32$  output features with  $K = 5$  filter taps. The nonlinearity function is ReLU in GNN and Lipschitz GNN. We regularize the Lipschitz continuity in Lipschitz GNN by adding a penalty term  $0.3h'(\lambda)$

to the loss function which is set as the cross-entropy loss. We use an ADAM optimizer with the learning-rate as 0.005 and the forgetting factors 0.9 and 0.999. We train each architecture for 40 epochs with the batch size set as 10. We average the estimation error rates by running 5 random dataset partitions.

**Convergence verification.** We first evaluate the convergence results by training the architectures on graphs with  $n = 300, 400, 500, 600, 700, 800, 900$  sampled points and comparing the graph outputs between the trained architectures on graphs with size  $n$  and on a large graph with size 1,000. The differences of the graph outputs is shown in Figure 2, where we can observe the convergence as the size of the trained graph grows. This proves our statement in Theorem 2 if we see the large enough graph as a good approximation of the underlying manifold. We can see that Lipschitz GNN performs better than GNN as continuous filter functions have better approximation and smaller convergence rate while GNN outperforms GF with the nonlinearity employed.

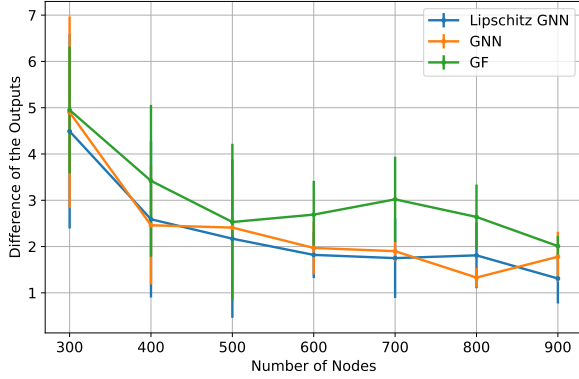


Fig. 2: Differences of the outputs of trained Lipschitz GNN, GNN and GF.

**Transferability verification.** We further justify the transferability by testing the trained architectures on a large graph with 1,000 sampled points. As Figure 4 shows, we train the architectures on graphs with fewer sampled points with  $n = 300, 500, 700, 900$  (Figure 3a and 4a) and directly implemented on graphs with a large set of sampled points (Figure 3b and 4b). Note that due to the non-scalability of the final linear readout layer, we only retrain the final linear layer while keeping the graph filter coefficients unchanged. The classification error rates are presented in Table I where we can observe that the trained architectures still can have good performances with Lipschitz GNN outperforms GNN while GNN is better than GF. This verifies that the filter continuity and nonlinearity can help improve the convergence and the transferability. Plus the architectures trained on graphs with more sampled points have better performances on the large graph. This is because architectures trained on larger graphs approximate better to manifold filters or MNNs.



(a) Trained graph (b) Tested graph

Fig. 3: Different sampled points on a chair model



(a) Trained graph (b) Tested graph

Fig. 4: Different sampled points on a chair model

	GF	GNN	Lipschitz GNN
$n = 300$	$19.83 \pm 5.94$	$7.74 \pm 4.05$	$7.68 \pm 3.75$
$n = 500$	$21.97 \pm 4.17$	$10.10 \pm 1.40$	$8.60 \pm 2.95$
$n = 700$	$13.85 \pm 3.81$	$7.45 \pm 4.03$	$8.02 \pm 2.77$
$n = 900$	$16.62 \pm 2.38$	$7.92 \pm 3.14$	$7.44 \pm 3.30$

TABLE I: Classification error rates (%) for model ‘chair’ when testing the architectures trained on sparse geometric graphs with  $n = 300, 500, 700, 900$  to sparse geometric graphs with  $n = 1,000$ . Average over 5 data realizations.

## VI. CONCLUSION

In this paper, we implement the definition of manifold convolutional filters with an integration of Laplace-Beltrami operator exponentials to process manifold signals. The manifold model is accessed by a set of i.i.d. uniformly sampled points over the manifold. We construct a relatively sparse graph to approximate the underlying manifold. With the approximation error bounds of discrete graph Laplacians to the LB operator in the spectral domain, we can prove the graph filter can approximate the manifold filter with a non-asymptotic error bound, which also provides the convergence rate of graph filters to the manifold filters. The approximation error bound shows a trade-off between the discriminability of the graph filters and the approximation (or convergence) to the manifold filters. GNNs made up of graph filters and nonlinearities can lift this trade-off due to the frequency mixing effects brought by nonlinearities. We conclude that the GNNs can therefore both converge to MNNs and discriminate the frequency components in high frequency domain well. We further analyze the transferability property of GNNs which allows a trained GNN directly implemented on another graph. We finally verify our convergence and transferability results numerically with a point-cloud classification problem.



## REFERENCES

- [1] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.
- [2] E. Tolstaya, J. Paulos, V. Kumar, and A. Ribeiro, "Multi-robot coverage and exploration using spatial graph neural networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8944–8950.
- [3] K. Atz, F. Grisoni, and G. Schneider, "Geometric deep learning on molecular representations," *Nature Machine Intelligence*, vol. 3, no. 12, pp. 1023–1032, 2021.
- [4] S. Li, J. Zhou, T. Xu, D. Dou, and H. Xiong, "Geomgcl: Geometric graph contrastive learning for molecular property prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4541–4549.
- [5] W. He, Z. Jiang, C. Zhang, and A. M. Sainju, "Curvanet: Geometric deep learning based on directional curvature for 3d shape analysis," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2214–2224.
- [6] J. Zeng, G. Cheung, M. Ng, J. Pang, and C. Yang, "3d point cloud denoising using graph laplacian regularization of a low dimensional manifold model," *IEEE Transactions on Image Processing*, vol. 29, pp. 3474–3489, 2019.
- [7] Z. Wang, L. Ruiz, M. Eisen, and A. Ribeiro, "Stable and transferable wireless resource allocation policies via manifold neural networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8912–8916.
- [8] R. Shelim and A. S. Ibrahim, "Geometric machine learning over riemannian manifolds for wireless link scheduling," *IEEE Access*, vol. 10, pp. 22 854–22 864, 2022.
- [9] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
- [10] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.
- [11] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [12] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2019.
- [13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [14] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.
- [15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [16] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5115–5124.
- [17] Z. Wang, L. Ruiz, and A. Ribeiro, "Stability to deformations of manifold filters and manifold neural networks," *arXiv preprint arXiv:2106.03725*, 2021.
- [18] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 37–45.
- [19] Z. Wang, L. Ruiz, and A. Ribeiro, "Convolutional neural networks on manifolds: From graphs and back," *arXiv preprint arXiv:2210.00376*, 2022.
- [20] M. Belkin and P. Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1289–1308, 2008.
- [21] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1711–1719.
- [22] M. Penrose, *Random geometric graphs*. OUP Oxford, 2003, vol. 5.
- [23] L. Ruiz, L. Chamon, and A. Ribeiro, "Graphon neural networks and the transferability of graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1702–1712, 2020.
- [24] N. Keriven, A. Bietti, and S. Vaiter, "Convergence and stability of graph convolutional networks on large random graphs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 512–21 523, 2020.
- [25] S. Maskey, R. Levie, and G. Kutyniok, "Transferability of graph neural networks: an extended graphon approach," *arXiv preprint arXiv:2109.10096*, 2021.
- [26] L. Ruiz, L. F. Chamon, and A. Ribeiro, "Transferability Properties of Graph Neural Networks," *arXiv preprint arXiv:2112.04629*, 2021.
- [27] J. Chew, D. Needell, and M. Perlmuter, "A convergence rate for manifold neural networks," *arXiv preprint arXiv:2212.12606*, 2022.
- [28] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [29] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.
- [30] R. Merris, "A survey of graph laplacians," *Linear and Multilinear Algebra*, vol. 39, no. 1-2, pp. 19–31, 1995.
- [31] J. Calder and N. G. Trillos, "Improved spectral convergence rates for graph laplacians on epsilon-graphs and k-nn graphs," *arXiv preprint arXiv:1910.13476*, 2019.
- [32] Y. Shi and B. Xu, "Gradient estimate of an eigenfunction on a compact riemannian manifold without boundary," *Annals of Global Analysis and Geometry*, vol. 38, no. 1, pp. 21–26, 2010.
- [33] A. Seelmann, "Notes on the  $\sin 2\theta$  theorem," *Integral Equations and Operator Theory*, vol. 79, no. 4, pp. 579–597, 2014.
- [34] W. Arendt and W. P. Schleich, *Mathematical analysis of evolution, information, and complexity*. John Wiley & Sons, 2009.
- [35] Z. Wang, L. Ruiz, and A. Ribeiro, "Stability of neural networks on manifolds to relative perturbations," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5473–5477.
- [36] —, "Convergence of graph neural networks on relatively sparse graphs. [Online]. Available: <https://zhiyangw.com/Papers/Asilomar2023.pdf>
- [37] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5680–5695, 2020.
- [38] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.
- [39] U. Von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *The Annals of Statistics*, pp. 555–586, 2008.

## APPENDIX

### A. Proof of Theorem 2

We first write out the filter representation as

$$\begin{aligned} & \| \mathbf{h}(\mathbf{L}_n^\epsilon) \mathbf{P}_n f - \mathbf{P}_n \mathbf{h}(\mathcal{L}) f \| \\ & \leq \left\| \sum_{i=1}^{\infty} \hat{h}(\lambda_{i,n}^\epsilon) \langle \mathbf{P}_n f, \phi_{i,n}^\epsilon \rangle \mathbf{G}_n \phi_{i,n}^\epsilon - \sum_{i=1}^{\infty} \hat{h}(\lambda_i) \langle f, \phi_i \rangle \mathcal{M} \mathbf{P}_n \phi_i \right\| \end{aligned} \quad (19)$$

We denote the index of partitions that contain a single eigenvalue as a set  $\mathcal{K}_s$  ( $|\mathcal{K}_s| = N_s$ ) and the rest as a set  $\mathcal{K}_m$  ( $|\mathcal{K}_m| = N_m$ ). We decompose the  $\alpha$ -FDT filter function as  $\hat{h}(\lambda) = h^{(0)}(\lambda) + \sum_{l \in \mathcal{K}_m} h^{(l)}(\lambda)$  as

$$h^{(0)}(\lambda) = \begin{cases} \hat{h}(\lambda) - \sum_{l \in \mathcal{K}_m} \hat{h}(C_l) & \lambda \in [\Lambda_k(\alpha)]_{k \in \mathcal{K}_s} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

$$h^{(l)}(\lambda) = \begin{cases} \hat{h}(C_l) & \lambda \in [\Lambda_k(\alpha)]_{k \in \mathcal{K}_s} \\ \hat{h}(\lambda) & \lambda \in \Lambda_l(\alpha) \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

with  $C_l$  some constant in  $\Lambda_l(\alpha)$ . With the triangle inequality, we start by analyzing the output difference of  $h^{(0)}(\lambda)$  as

$$\begin{aligned} & \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_{i,n}^{\epsilon}) \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} - \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \langle f, \phi_i \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right\| \\ & \leq \left\| \sum_{i=1}^{\infty} \left( h^{(0)}(\lambda_{i,n}^{\epsilon}) - h^{(0)}(\lambda_i) \right) \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} \right\| \\ & + \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} - \langle f, \phi_i \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right) \right\|. \end{aligned} \quad (22)$$

The first term in (22) can be bounded by leveraging the  $A_h$ -Lipschitz continuity of the frequency response. From the eigenvalue difference in Proposition 2, we can claim that for each eigenvalue  $\lambda_i \leq \lambda_K$ , we have

$$|\lambda_{i,n}^{\epsilon} - \lambda_i| \leq C_{K,1} \sqrt{\epsilon}. \quad (23)$$

The square of the first term is bounded as

$$\begin{aligned} & \left\| \sum_{i=1}^{\infty} (h^{(0)}(\lambda_{i,n}^{\epsilon}) - h^{(0)}(\lambda_i)) \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} \right\|^2 \\ & \leq \sum_{i=1}^{\infty} |h^{(0)}(\lambda_{i,n}^{\epsilon}) - h^{(0)}(\lambda_i)|^2 |\langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n}|^2 \quad (24) \\ & \leq \sum_{i=1}^{\infty} A_h^2 |\lambda_{i,n}^{\epsilon} - \lambda_i|^2 \|\mathbf{P}_n f\|^2 \leq A_h^2 C_{K,1}^2 \epsilon. \end{aligned} \quad (25)$$

The second term in (22) can be bounded combined with the convergence of eigenfunctions in (27) as

$$\begin{aligned} & \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} - \langle f, \phi_i \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right) \right\| \\ & \leq \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} - \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \mathbf{P}_n \phi_i \right) \right\| \\ & + \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \mathbf{P}_n \phi_i - \langle f, \phi_i \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right) \right\| \end{aligned} \quad (26)$$

From the convergence stated in Theorem 2, we have

$$\|a_i \phi_{i,n}^{\epsilon} - \phi_i\| \leq C_{K,2} \sqrt{\epsilon} / \theta, \quad (27)$$

with the eigengap  $\theta \geq \alpha$  under the  $\alpha$ -FDT filter. Therefore, the first term in (26) can be bounded as

$$\begin{aligned} & \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_i) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \phi_{i,n}^{\epsilon} - \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right) \right\| \\ & \leq \sum_{i=1}^{N_s} \|\mathbf{P}_n f\| \|\phi_{i,n}^{\epsilon} - \mathbf{P}_n \phi_i\| \leq \frac{N_s C_{K,2}}{\alpha} \sqrt{\epsilon}. \end{aligned} \quad (28)$$

The last equation comes from the definition of norm in  $L^2(\mathbf{G}_n)$ .

The second term in (26) can be written as

$$\begin{aligned} & \left\| \sum_{i=1}^{\infty} h^{(0)}(\lambda_{i,n}^{\epsilon}) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} \mathbf{P}_n \phi_i - \langle f, \phi_i \rangle_{\mathcal{M}} \mathbf{P}_n \phi_i \right) \right\| \\ & \leq \sum_{i=1}^{\infty} |h^{(0)}(\lambda_{i,n}^{\epsilon})| \left| \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} - \langle f, \phi_i \rangle_{\mathcal{M}} \right| \|\mathbf{P}_n \phi_i\|. \end{aligned} \quad (29)$$

Because  $\{x_1, x_2, \dots, x_n\}$  is a set of uniform sampled points from  $\mathcal{M}$ , based on Theorem 19 in [39] we can claim that

$$|\langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} - \langle f, \phi_i \rangle_{\mathcal{M}}| = O\left(\sqrt{\frac{\log n}{n}}\right). \quad (30)$$

Taking into consider the boundedness of frequency response  $|h^{(0)}(\lambda)| \leq 1$  and the bounded energy  $\|\mathbf{P}_n \phi_i\|$ . Therefore, we have

$$\left\| \sum_{i=1}^{\infty} \hat{h}(\lambda_{i,n}^{\epsilon}) \left( \langle \mathbf{P}_n f, \phi_{i,n}^{\epsilon} \rangle_{\mathbf{G}_n} - \langle f, \phi_i \rangle_{\mathcal{M}} \right) \mathbf{P}_n \phi_i \right\| = O\left(\sqrt{\frac{\log n}{n}}\right).$$

Combining the above results, we can bound the output difference of  $h^{(0)}$ . Then we need to analyze the output difference of  $h^{(l)}(\lambda)$  and bound this as

$$\begin{aligned} & \left\| \mathbf{P}_n \mathbf{h}^{(l)}(\mathcal{L}) f - \mathbf{h}^{(l)}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f \right\| \\ & \leq \left\| (\hat{h}(C_l) + \gamma) \mathbf{P}_n f - (\hat{h}(C_l) - \gamma) \mathbf{P}_n f \right\| \leq 2\gamma \|\mathbf{P}_n f\|, \end{aligned} \quad (31)$$

where  $\mathbf{h}^{(l)}(\mathcal{L})$  and  $\mathbf{h}^{(l)}(\mathbf{L}_n^{\epsilon})$  are filters with filter function  $h^{(l)}(\lambda)$  on the LB operator  $\mathcal{L}$  and graph Laplacian  $\mathbf{L}_n^{\epsilon}$  respectively. Combining the filter functions, we can write

$$\begin{aligned} & \left\| \mathbf{P}_n \mathbf{h}(\mathcal{L}) f - \mathbf{h}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f \right\| \\ & = \left\| \mathbf{P}_n \mathbf{h}^{(0)}(\mathcal{L}) f + \mathbf{P}_n \sum_{l \in \mathcal{K}_m} \mathbf{h}^{(l)}(\mathcal{L}) f - \mathbf{h}^{(0)}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f - \sum_{l \in \mathcal{K}_m} \mathbf{h}^{(l)}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f \right\| \end{aligned} \quad (32)$$

$$\begin{aligned} & \leq \left\| \mathbf{P}_n \mathbf{h}^{(0)}(\mathcal{L}) f - \mathbf{h}^{(0)}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f \right\| + \\ & \sum_{l \in \mathcal{K}_m} \left\| \mathbf{P}_n \mathbf{h}^{(l)}(\mathcal{L}) f - \mathbf{h}^{(l)}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f \right\| \end{aligned} \quad (33)$$

$$\leq A_h C_{K,1} \sqrt{\epsilon} + N_s C_{K,2} \sqrt{\epsilon} + N_m \gamma + C_{gc} \sqrt{\frac{\log(n)}{n}}$$

$$\begin{aligned} & \left\| \mathbf{h}(\mathbf{L}_n^{\epsilon}) \mathbf{P}_n f - \mathbf{P}_n \mathbf{h}(\mathcal{L}) f \right\| \\ & \leq \left( \frac{N C_{K,2}}{\alpha} + A_h C_{K,1} \right) \sqrt{\epsilon} + C_{gc} \sqrt{\frac{\log n}{n}} \end{aligned} \quad (34)$$