Transferability of Spectral Graph Convolutional Neural Networks

Ron Levie[*], **Wei Huang**[†], **Lorenzo Bucci**[†], **Michael Bronstein**[†§‡], **Gitta Kutyniok**[*¶]

$*$ Technische Universität Berlin, †Università della Svizzera italiana,
¶University of Tromsø , ‡Imperial College London, §Twitter

Abstract

This paper focuses on spectral graph convolutional neural networks (CNNs), where filters are defined as elementwise multiplication in the frequency domain of a graph. In machine learning settings where the dataset consists of signals defined on many different graphs, the trained CNN should generalize to signals on graphs unseen in the training set. It is thus important to transfer filters from one graph to the other. Transferability, which is a certain type of generalization capability, can be loosely defined as follows: if two graphs represent the same phenomenon, then a single filter/CNN should have similar repercussions on both graphs. This paper aims at debunking the common misconception that spectral filters are not transferable. We show that if two graphs are discretizations of the same underlying space, then a spectral filter/CNN has approximately the same repercussion on both graphs. Our analysis is more permissive than the standard analysis. Transferability is typically described as the robustness of the filter to small graph perturbations and re-indexing of the vertices. We prove transferability between graphs that can have completely different dimensions and topologies, only requiring that both graphs discretize the same underlying space.

1 Introduction

The success of convolutional neural networks on Euclidean domains ignited an interest in recent years in extending these methods to graph structured data. In a standard CNN, the network receives as input a signal defined over a Euclidean rectangle, and at each layer applies a set of convolutions/filters on the outputs of the previous layer, a non linear activation function, and, optionally, pooling. A graph CNN has the same architecture, with the only difference that now signals are defined over the vertices of graph domains, and not Euclidean rectangles. Graph structured data is ubiquitous in a range of application, and can represent 3D shapes, molecules, social networks, point clouds, and citation networks to name a few. In some situations, the data consists of many different graphs, and many different signals on these graphs (multi-graph setting). In this situation, if two graphs represent the same underlying phenomenon, and the two signals given on the two graphs are similar in some sense, the output of the CNN on both signals should be similar as well. This property is typically termed transferability, and is an essential requirement if we wish the CNN to generalize well on the test set in multi-graph settings. In fact, transferability can be seen as special type of generalization capability. Analyzing and proving transferability of spectral graph CNNs is the focus of this paper.

Graph CNNs can manage transferability in different ways. First, when a graph CNN is shown a multi-graph training set, it can learn "concepts" that promote transferability. Let us call this approach *concept-based transferability*. Second, it may be the case that transferability is a mathematical law: a built-in capability of certain types of graph CNNs, independent of their specific filters, which requires no training. This approach, that we call *principle transferability*, is the focus of this paper.

We believe that the success of spectral graph CNNs in multi-graph settings relies on both types of transferability. We call the accumulative effect of concept-based transferability and principle transferability *total transferability*. In this paper we prove theoretically that spectral graph CNNs

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

have principle transferability. We moreover demonstrate principle transferability by concocting experiments that isolate principle transferability from concept-based transferability. This is done by training the network on one single graph, which prevents it from learning concepts for dealing with varying graphs, and testing the resulting network on other graphs. The performance of such a network on the new graphs only partially degrades, illustrating the effect size of principle transferability in total transferability. Moreover, in this isolated principle transferability experiment, spectral methods do not perform worse than spatial methods, which indicates that spectral methods have competitive transferability capabilities.

2 Convolution operators on graphs

There are generally two approaches to defining convolution operators on graphs, both generalizing the standard convolution on Euclidean domains [2, 13]. Spatial approaches generalize the idea of a sliding window to graphs. Here, the main challenge is to define a way to aggregate feature information from the neighbors of each node. Some popular examples of spatial methods are [7, 12, 10]. Spectral methods are inspired by the convolution theorem in Euclidean domains, that states that convolution in the spatial domain is equivalent to pointwise multiplication in the frequency domain. To define the frequency domain of a graph, we consider the graph Laplacian $\mathbf{\Delta}$ or some other graph shift operator, and use its eigenvalues as frequencies and its eigenvectors as the corresponding Fourier modes [11]. The filter $\mathbf{F}$ is defined on the graph signal $\mathbf{s}$ by

$$\mathbf{F}\mathbf{s} = \sum_{n=1}^{N} f_n(\boldsymbol{\psi}_n^* \cdot \mathbf{s})\boldsymbol{\psi}_n \tag{1}$$

where $\{\boldsymbol{\psi}_n\}_{n=1}^N$ are the eigenvectors of $\mathbf{\Delta}$, and $\boldsymbol{\psi}_n^*$ is the conjugate transpose of $\boldsymbol{\psi}_n$. Here, the scalars $\{f_n\}_{n=1}^N$ are the frequency responses of the filter, and different choices of f_n results in different types of filters, e.g., high, middle and low pass filters. For some examples of spectral methods see, e.g., [3, 4, 9, 6].

State of the art spectral graph CNNs are not based on the naive implementation (1) of graph filters, but rather on a *functional calculus* implementation. As oppose to (1), where the frequency responses are parametrized by the *index* n of the eigenvectors, the frequency response in functional calculus filters is parametrized by the *value* λ_n of the eigenvalues corresponding to the eigenvecors $\boldsymbol{\psi}_n$. Namely, we define filters by

$$\mathbf{F}\mathbf{s} = f(\mathbf{\Delta})\mathbf{s} := \sum_{n=1}^{N} f(\lambda_n)(\boldsymbol{\psi}_n^* \cdot \mathbf{s})\boldsymbol{\psi}_n \tag{2}$$

where the frequency response $f : \mathbb{R} \to \mathbb{C}$ is now a function. It turns out that spectral filters based on functional calculus implementation are linearly stable with respect to perturbations in the graph Laplacian [8, 5]. Moreover, functional calculus filters are computationally efficient [4, 9].

One typical motivation for favoring spatial methods is the claim that spectral methods are not transferable, and thus are not appropriate in multi-graph settings. The goal in this paper is to debunk this misconception, and to show that state-of-the-art spectral graph filtering methods are transferable. Interestingly, [1] obtained state-of-the-art results using spectral graph filters in a multi-graph setting, without any modification to compensate for the "non-transferability".

3 Principle transferability of spectral graph CNNs

We present a framework of transferability, allowing to compare graphs of incompatible sizes and topologies. We consider spectral filters as they are, and do not enhance them with any computational machinery for transferring filters. Thus, one of the main conceptual challenges is to find a way to compare two different graphs, with incompatible graph structures, from a theoretical stance. To accommodate the comparison of incompatible graphs, our approach resorts to non-graph theoretical considerations, assuming that graphs are observed from some underlying spaces that may or may not be graphs. One example is when graphs are regarded as discretizations of underlying corresponding continuous metric spaces. This makes sense, since a weighted graph can be interpreted as a set of points (vertices) and a decreasing function of their distances (edge weights). Two graphs are comparable, or represent the same phenomenon, if both discretize the same space. This approach allows us to prove transferability under small perturbations of the adjacency matrix, but more generally, allows us to prove transferability between graphs with incompatible structures.

The way to compare two graphs is to consider their embeddings to the space they both discretize. For intuition, consider the special case where the metric space is a manifold. Any manifold can be discretized to a graph/polygon-mesh in many ways, resulting in different graph topologies. A filter designed/learned on one polygon-mesh should have approximately the same repercussion on a different polygon-mesh discretizing the same manifold. To compare the filter on the two graphs, we consider a generic signal defined on the continuous space, and sampled to both graphs. After applying the graph filter on the sampled signal on both graphs, we interpolate both results back to two continuous signals. In our analysis we show that these two interpolated continuous signals are approximately equal.

Consider a metric space $\mathcal{M}$ with a Borel measure μ, and the space of signals $L_2(\mathcal{M})$. In general, the space $L_2(\mathcal{M})$ may be infinite dimensional. Consider a self-adjoint operator $\mathcal{L}$ in $L_2(\mathcal{M})$ that we call the *metric Laplacian*. We suppose that $\mathcal{L}$ has a discrete spectrum, with eigenvalues $\lambda_0 < \lambda_1 < \ldots$, and corresponding eigenfunctions $\phi_n : \mathcal{M} \to \mathbb{C}$. The metric-Laplacian models the geometry in $\mathcal{M}$, and specifically formalizes the notion of oscillations or frequencies. We define band-limited spaces in $L_2(\mathcal{M})$, also called Paley-Wiener spaces, by

$$PW(\lambda_M) = \mathrm{span}\{\phi_m\}_{m=0}^{M}.$$

Denote by $P(\lambda_M)$ the orthogonal projection upon $PW(\lambda_M)$. Graphs are sampled from metric spaces using sampling operators. In general, given the graph G, we define sampling as a linear operator $S : \mathcal{C}(\mathcal{M}) \to L^2(G)$, where $\mathcal{C}(\mathcal{M})$ is the space of continuous functions in $\mathcal{M}$. We define for each Paley-Wiener space $P(\lambda_M)$ an interpolation operator as a linear operator $R_\lambda : L^2(G) \to PW(\lambda_M)$. We consider a graph Laplacian $\mathbf{\Delta}$ of G that may be any self-adjoint shift operator.

Our theory applies on any model of S, R_λ, which are only required to be linear and bounded. For a concrete example, consider sampling by evaluation on sample points in a metric space $\mathcal{M}$. Given N sample points $G = \{x_k\}_{k=1}^N \subset \mathcal{M}$, the sampling operator $S : C(\mathcal{M}) \to L_2(G)$ is defined by

$$Sq = \{q(x_k)\}_{k=1}^{N_n}.$$

Given a graph signal $\mathbf{q} \in L_2(G)$, we define the interpolation of $\mathbf{q}$ to the Paley-Wiener space $PW(\lambda_M)$ as the adjoint operator of the operator $SP(\lambda_M)$, namely, $R_{\lambda_M} = \big(SP(\lambda_M)\big)^*$.

In another example, $\mathcal{M}$ is a graph, and G is a coarsened version of $\mathcal{M}$, where pairs of neighboring nodes in $\mathcal{M}$ are collapsed to single nodes in G. Sampling is the operator that assigns the value $2^{-1/2}q_1 + 2^{-1/2}q_2$ to the node of G with parent nodes from $\mathcal{M}$ have the values q_1 and q_2. Interpolation is defined to be $R = S^*$ on the whole spectral gap $\lambda_{\max}$. In a simpler example, $\mathcal{M}$ is a graph, and G is a perturbation of $\mathcal{M}$, that is obtain by adding/deleting random edges from $\mathcal{M}$ or perturbing the edge weights. Here, $S = R = I$.

We formulate transferability between filters in $\mathcal{M}$ and filters in G has follows

Theorem 1 *Consider the above setting, and let $\lambda_M > 0$ be a band with $\|R_{\lambda_M}\| < C$. Let $f : \mathbb{R} \to \mathbb{C}$ be a Lipschitz continuous function, with Lipschitz constant D, and denote $\|f\|_{\mathcal{L},M} = \max_{0 \leq m \leq M}\{|f(\lambda_m)|\}$. Then the following two bounds are satisfied*

$$\begin{aligned} \|f(\mathcal{L})P(\lambda_M) - R_{\lambda_M} f(\mathbf{\Delta}_n) S_n P(\lambda_M)\| \leq & DC\sqrt{M}\, \|S\mathcal{L}P(\lambda_M) - \mathbf{\Delta} S P(\lambda_M)\| \\ & + \|f\|_{\mathcal{L},M} \|P(\lambda_M) - R_{\lambda_M} S_n P(\lambda_M)\| . \end{aligned} \quad (3)$$

$$\|f(\mathcal{L})q - R_{\lambda_M} f(\mathbf{\Delta}_n) S_n q\| \leq DC \sum_{m=0}^{M} |c_m| \, \|S\mathcal{L}\phi_m - \mathbf{\Delta} S\phi_m\| + \|f\|_{\mathcal{L},M} \|q - R_{\lambda_M} S_n q\| . \quad (4)$$

where $q = \sum_{m=0}^{M} c_m \phi_m$.

To interpret (3) and (4), the left-hand-side is the transferability error in the filter. The first term in the right-hand-side is the transferability error in the Laplacian, either in maximal spectral norm or applied on the different Fourier modes, and the second term is the error entailed by sampling-interpolation a signal.

Now, consider two graphs G_1 and G_2, with corresponding graph Laplacians $\mathbf{\Delta}_1$ and $\mathbf{\Delta}_2$, that represent the same phenomenon. Adopting our basic assumption, we thus suppose that both graphs approximate the space $\mathcal{M}$ in the following sense. For some fixed Paley-Wiener space $PW(\lambda_M)$, and for each $n = 1, 2$ and any signal $q \in PW(\lambda_M)$, we suppose $\|\mathcal{L}q - R_{n;\lambda_M}\mathbf{\Delta}_n S_n q\| \approx 0$ and $\|q - R_{n;\lambda_M} S_n q\| \approx 0$. By the triangle inequality, we can also show

$$\|R_{1;\lambda_M}\mathbf{\Delta}_1 S_1 q - R_{2;\lambda_M}\mathbf{\Delta}_2 S_2 q\|\,, \quad \|R_{1;\lambda_M} S_1 q - R_{2;\lambda_M} S_2 q\| \approx 0. \tag{5}$$

As a result of (3) and by the triangle inequality, we have

$$\begin{aligned} &\|R_{1;\lambda_M} f(\mathbf{\Delta}_1) S_1 q - R_{2;\lambda_M} f(\mathbf{\Delta}_2) S_2 q\| \\ &= O\Big(\|R_{1;\lambda_M}\mathbf{\Delta}_1 S_1 q - R_{2;\lambda_M}\mathbf{\Delta}_2 S_2 q\| + \max_{n=1,2} \|q - R_{n;\lambda_M} S_n q\| \Big). \end{aligned} \tag{6}$$

4 Experiments

In top-left of Figure 1 we isolate principle transferability form concept-based transferability in MNIST, and compare a spectral graph CNN method with a spatial graph CNN method. We consider a simple CNN architecture based on two convolution layers, where the max pooling in the second layer collapses each channel to one node, and two fully connected layers. Both the spectral and spatial methods have 92K parameters. We train the network on MNIST images of one fixed fine resolution ($32X32$) and test on images of various coarse resolutions. The graph Laplacian is given by the central difference approximating second derivative. In this setting, the spectral method, CayleyNet, has higher principle transferability than the spatial method, MoNet. Indeed, its performance degrades slower as we coarsen the grid.

In top-middle and right of Figure 1 we test transferability between the Citeseer graph $\mathcal{M}$ and its coarsened version G as described in Section 3. We consider the coarsening and interpolation operators S and $R = S^*$ of Section 3. We consider the normalized Laplacian $\mathcal{L}$ on $\mathcal{M}$, and the coarse Laplacian $\mathbf{\Delta} = S\mathcal{L}R$ on G. We consider low-pass (top-middle) and high-pass (top-right) filters with Lipschitz constant 1. To show transferability, we plot $\|Sf(\mathcal{L})\phi_m - f(\mathbf{\Delta})S\phi_m\|$ as a function of $\|S\mathcal{L}\phi_m - \mathbf{\Delta}S\phi_m\|$ for various eigenvectors ϕ_m of $\mathcal{L}$ (some corresponding eigenvalues are displayed). All values lie below $y = x$, which accord with (1) of the supplementary material.

In Figure 1 bottom, we test the stability of spectral graph filters in the Cora graph with the normalized Laplacian, for different models of graph perturbations and sub-sampling. We consider three filters: low, mid and high pass. In bottom-left we randomly remove edges, in bottom-middle we randomly add edges, and in bottom-right randomly delete vertices, and compare the filters on the sub-sampled graph. The markers indicate the percentage of edges/vertices that were removed/added. The x axis is the relative error in the Laplacian, and the y axis is the relative error in the filter. The experimental results support the theoretical results on linear stability. All errors are given in Frobenius norm. The Frobenius norm can be seen as the average pointwise error, where the Laplacians and filters are applied on the signals of the standard basis.

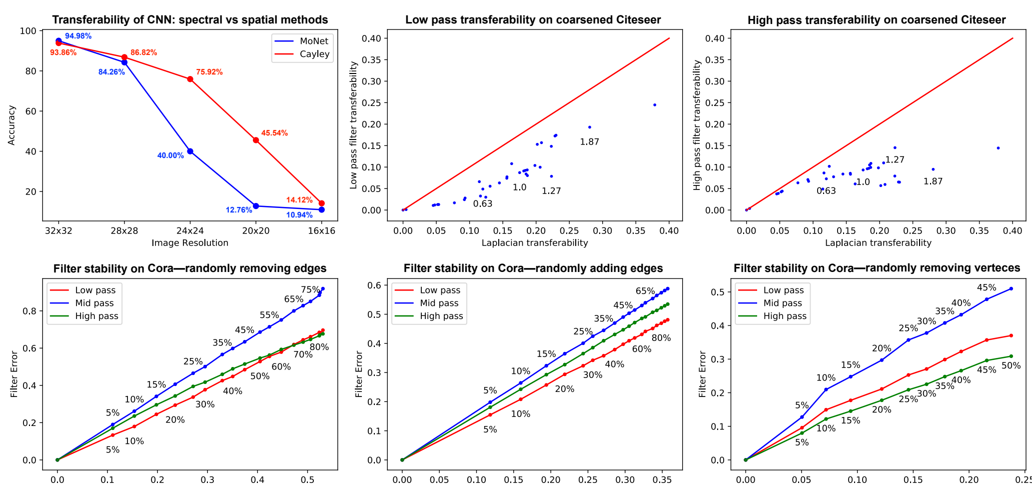

Figure 1: Trasferability experiments

References

[1] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi. Graph neural networks with convolutional arma filters. *arXiv:1901.01343, preprint*, 2019.

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017.

[3] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *Proc. ICLR*, 2013.

[4] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. NIPS*, 2016.

[5] F. Gama, J. Bruna, and A. Ribeiro. Stability properties of graph neural networks. *arXiv:1905.04497, preprint*, 2019.

[6] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro. Convolutional neural networks architectures for signals supported on graphs. *arXiv:1805.00165, preprint*, 2018.

[7] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, July 2005.

[8] R. Levie, E. Isufi, and G. Kutyniok. On the transferability of spectral graph filters. *arXiv preprint arXiv:1901.10524*, 2019.

[9] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, Jan 2019.

[10] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5425–5434, 2017.

[11] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, Jan 2009.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv:1901.00596, preprint*, 2019.

Transferability of Spectral Graph Convolutional Neural Networks Supplementary Material

Ron Levie*, **Wei Huang**†, **Lorenzo Bucci**†, **Michael Bronstein**†§‡, **Gitta Kutyniok***¶

∗ Technische Universität Berlin, †Università della Svizzera italiana,
¶University of Tromsø , ‡Imperial College London, §Twitter

A Proof of Theorem 1

By linearity and finite dimension of $PW(\lambda)$, we start a signal $s = \phi_m$ which is an eigenvector of $\mathcal{L}$ corresponding to the eigenvalue λ_j, and then generalize to linear combinations. Let P_k^n be the projection upon the eigenspace of $\mathbf{\Delta}_n$ corresponding to the eigenvalues κ_k^n of $\mathbf{\Delta}_n$. Then,

$$\begin{aligned}\mathbf{\Delta}_n S_n\phi_m - S_n\mathcal{L}\phi_m &= \sum_k \kappa_k^n P_k^n S_n\phi_m - \lambda_m S_n\phi_m \\ &= \sum_k (\kappa_k^n - \lambda_m) P_k^n S_n\phi_m.\end{aligned}$$

By orthogonality of the projections $\{P_k^n\}_k$,

$$\left\| \sum_k \kappa_k^n P_k^n S_n\phi_m - \lambda_m S_n\phi_m \right\|^2 = \sum_k |\kappa_k^n - \lambda_m|^2 \left\| P_k^n S_n\phi_m \right\|^2$$

Now, since f is Lipschitz,

$$\begin{aligned}\left\| f(\mathbf{\Delta}_n)S_n\phi_m - S_n f(\mathcal{L})\phi_m \right\|^2 =& \sum_k |f(\kappa_k^n) - f(\lambda_j)|^2 \left\| P_k^n S_n\phi_m \right\|^2 \\ \leq& D^2 \sum_k |\kappa_k^n - \lambda_m|^2 \left\| P_k^n S_n\phi_m \right\|^2 \\ =& D^2 \left\| \sum_k \kappa_k^n P_k^n S_n\phi_m - \lambda_m S_n\phi_m \right\|^2 \\ =& D^2 \left\| \mathbf{\Delta}_n S_n\phi_m - S_n\mathcal{L}\phi_m \right\|^2 .\end{aligned} \tag{1}$$

Now, any $s \in L^2(\mathcal{M})$ can be written as

$$s = \sum_m c_m\phi_m.$$

We have

$$\left\| f(\mathbf{\Delta}_n)S_n P(\lambda_M)s - S_n f(\mathcal{L})P(\lambda_M)s \right\| = \left\| \sum_{m=1}^M c_m \Big(f(\mathbf{\Delta}_n)S_n - S_n f(\mathcal{L}) \Big)\phi_m \right\|.$$

By the triangle inequality,

$$\begin{aligned}\left\| f(\mathbf{\Delta}_n)S_n P(\lambda_M)s - S_n f(\mathcal{L})P(\lambda_M)s \right\| &\leq \sum_{m=1}^M |c_m| \left\| \Big(f(\mathbf{\Delta}_n)S_n\phi_m - S_n f(\mathcal{L})\phi_m \Big) \right\| \\ &\leq \left\| s \right\|_1 D \left\| \mathbf{\Delta}_n S_n P(\lambda_M) - S_n\mathcal{L}P(\lambda_M) \right\| .\end{aligned} \tag{2}$$

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Here, $\|s\|_1 := \sum_{m=1}^{M} |c_m|$, satisfies

$$\|s\|_1 \leq \|s\|_2 \sqrt{M}.$$

Let us now derive another bound. Denote the vector

$$\mathbf{E} = \{E_m\}_m = \{\left\| \Big(f(\mathbf{\Delta}_n) S_n \phi_m - S_n f(\mathcal{L}) \phi_m \Big) \right\| \}_m.$$

The vector $\mathbf{E}$ represents the Laplacian error on the Fourier modes. Then

$$\begin{aligned} \|f(\mathbf{\Delta}_n) S_n P(\lambda_M) s - S_n f(\mathcal{L}) P(\lambda_M) s\| &\leq \sum_{m=1}^{M} |c_m| \left\| \Big(f(\mathbf{\Delta}_n) S_n \phi_m - S_n f(\mathcal{L}) \phi_m \Big) \right\| \\ &\leq \sum_{m=1}^{M} |c_m| \, D \, \|\mathbf{\Delta}_n S_n P(\lambda_M) \phi_m - S_n \mathcal{L} P(\lambda_M) \phi_m\| \end{aligned} \tag{3}$$

Last, by the triangle inequality on (2),

$$\begin{aligned} &\|f(\mathcal{L}) P(\lambda_M) - I_{n;\lambda_M} f(\mathbf{\Delta}_n) S_n P(\lambda_M)\| \\ &\leq \|f(\mathcal{L}) P(\lambda_M) - I_{n;\lambda_M} S_n f(\mathcal{L}) P(\lambda_M)\| + \|I_{n;\lambda_M} S_n f(\mathcal{L}) P(\lambda_M) - I_{n;\lambda_M} f(\mathbf{\Delta}_n) S_n P(\lambda_M)\| \\ &\leq \|P(\lambda_M) - I_{n;\lambda_M} S_n P(\lambda_M)\| \, \|f(\mathcal{L}) P(\lambda_M)\| + \|I_{n;\lambda_M}\| \, \|S_n f(\mathcal{L}) P(\lambda_M) - f(\mathbf{\Delta}_n) S_n P(\lambda_M)\| \, . \end{aligned}$$

Note that by assumption

$$\|I_{n;\lambda_M}\| \leq C$$

and, by the diagonal form of $f(\mathcal{L}) P(\lambda_M)$,

$$\|f(\mathcal{L}) P(\lambda_M)\| \leq \|f\|_{\mathcal{L},M} \, ,$$

which gives (4) of Theorem 1. A similar argument based on (3) gives (5) of Theorem 1.