LAPLACE-TRANSFORM-FILTERS RENDER SPECTRAL GRAPH NEURAL NETWORKS TRANSFERABLE

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a new point of view on transferability of graph neural networks based on the intrinsic notion of information diffusion within graphs. This notion is adapted to considering graphs to be similar if their overall rough structures are similar, while their fine-print articulation may differ. Transferability of graph neural networks is then considered between graphs that are similar from this novel perspective on transferability. After carefully analysing transferability of single filters, the transferability properties of entire networks are relegated to the transferability characteristics of the filters employed inside their convolutional blocks. A rigorous analysis establishes our main theoretical finding: Spectral convolutional networks are transferable between graphs whose overall rough structures align, if their filters arise as Laplace transforms of certain generalized functions. Numerical experiments illustrate and validate the theoretical findings in practice.

1 INTRODUCTION

A fundamental quality of any machine learning model is its ability to generalize beyond the data on which it was trained. In the graph neural network (GNN) setting, a crucial aspect of this capability is characterized by the property of transferability: If two graphs are similar, also their respective latent embeddings should be similar to each other. I.e. GNNs should be *transferable* between such graphs.

We may thus think of transferability as encoding information about continuity properties of GNNs: Equipping the space of graphs, with a suitable distance-notion capturing graph similarity, we may consider GNNs as functions mapping from this space to latent Euclidean spaces. Transferable models then correspond to continuous maps: Their outputs are close if input graphs are close to each other. In contrast, non-transferable GNNs are discontinuous: Embeddings generated by such models may vary strongly even if the corresponding graphs are close to each other. If a transferable GNN model is then confronted during inference with a graph that is similar to a graph that was already observed during training, generated latent embeddings will be similar. Hence a good performance on the train-set will translate to a similarly good performance on the test set: The model will be able to generalize.

Here we will be analyzing transferability properties of spectral graph neural networks (Bruna et al., 2014; Defferrard et al., 2016); a prominent class of GNNs which continue to set the state of the art on a diverse set of tasks (He et al., 2021; 2022a; Wang & Zhang, 2022; Koke & Cremers, 2024). From a theoretical perspective, transferability of such models has been predominantly investigated in the setting of (very) large graphs taken to faithfully approximate a common underlying ambient object. Examples of such objects are metric measure spaces (Levie et al., 2019a) and graphons (Ruiz et al., 2020; Maskey et al., 2021), which are applicable to graphs where the number of edges $|\mathcal{E}|$ is of $\mathcal{O}(N^2)$, with N the number of nodes. Large sparse graphs ($|\mathcal{E}| = \mathcal{O}(N)$) are instead considered to approximate the same graphop (Le & Jegelka, 2023) or graphing (Roddenberry et al., 2022). Transferability outside this asymptotic regime of large graphs has to the best of our knowledge so far only been investigated for limited examples and a restrictive class of filter functions Koke (2023).

Contributions: Here we propose an alternative approach to transferability: Fundamentally, we consider two graphs to be similar if the rough overall structures within them align, while fine-print articulations are allowed to vary. This setting captures fundamental examples such as graphs discretizing the same manifold, graphs describing the same object at different resolutions or graphs differing by edge deletions. To quantify similarity in this setting, we build on the notion of diffusion distance (Hammond et al., 2013), which provides a relaxation of the canonical linear distance $||L - \tilde{L}||$

between Laplacians $L, \tilde{L}$ of different graphs. Within this relaxed distance measure, variations in coarse structure are weighted more heavily, while variations in fine-structure are instead discounted. A rigorous analysis then establishes our main theoretical finding: Networks are transferable between graphs that are close in the diffusion sense, if their filters arise as Laplace transforms.

Our novel viewpoint provides a broad and general framework to analyze transferability: It is not dependent on any ambient space, applies outside the setting of large graphs, is not restricted to a certain scaling behaviour of the number of edges and covers settings where previous transferability results are not applicable (e.g. between original and coarsified graphs). To provide guidance for the practicioner, we perform carefully designed numerical experiments highlighting the importance of transferability, showcasing the failure of common architectures to transfer and numerically verifing that architectures conforming to our developed theory indeed do exhibit transferability.

Caveat: The notion of diffusion-similarity central to our analysis below is adapted to the setting where the rough overall structure within graphs is more important than fine structure details. Utilizing such a *relaxation* of the standard linear distance $||L - \tilde{L}||$ allows to consider more relaxed conditions on filter functions than previous works (Gama et al., 2019; Wang et al., 2021) in this setting. It is however important to note that since our analysis is based on a distance notion that discounts fine-structure details within graphs, the results in our paper do not allow to draw conclusions about transferability and model performance in settings where the exact articulation of a graph is important.

2 Background: Spectral Convolutional Networks on Graphs

2.1 Graphs and their fundamental Properties

Graphs: A graph $G := (\mathcal{G}, \mathcal{E})$ is a collection of nodes $\mathcal{G}$ and edges $\mathcal{E} \subseteq \mathcal{G} \times \mathcal{G}$. We assume (real) edge-weights with potentially $A_{ij} \neq A_{ji}$ if the graph is directed. Nodes $i \in \mathcal{G}$ may have individual node-weights $\mu_i > 0$. In a social network, a node weight $\mu_i = 1$ might e.g. signify that node i represents a single user. A weight $\mu_j > 1$ would indicate that node j represents a group of users.

Feature spaces: Given F-dimensional node features on a graph with $N = |\mathcal{G}|$ nodes, we collect individual scalar node-signals $x \in \mathbb{R}^N$ into a feature matrix X of dimension $N \times F$. Taking node weights into account, we equip the space of such signals with an inner-product according to $\langle X, Y \rangle = \text{Tr}(X^\intercal MY) = \sum_{i=1}^{N} \sum_{j=1}^{F} (\overline{X}_{ij} Y_{ij})\mu_i$ with $M = \text{diag}\left(\{\mu_i\}\right)$ the node-weight matrix.

Graph Laplacians: Spectral graph neural networks are typically based on some choice of (positive semi-definite) graph Laplacian L (Defferrard et al., 2016; He et al., 2021; 2022b), on which we will hence also focus here. Most important to us will be the un-normalized (in-degree) graph Laplacian $L = M^{-1}(D - A)$. Here A is the (weighted) adjacency matrix and D is the diagonal degree matrix.

2.2 Spectral Convolutional Filters

A spectral graph convolutional filter is then constructed by applying a learnable function $h_\theta(\cdot)$ to an underlying characteristic operator L; typically a graph Laplacian. The resulting filter matrix $h_\theta(L) \in \mathbb{R}^{N \times N}$ acts on scalar graph signals $x \in \mathbb{R}^N$ via matrix multiplication; sending x to $h_\theta(L) \cdot x$:

$$x \mapsto h_\theta(L) \cdot x$$

In practice it is prohibitively expensive to implement such filters using e.g. an explicit eigendecomposition (Defferrard et al., 2016). Instead, a generic filter function $h_\theta(\cdot)$ is typically parameterized as a weighted sum over 'simpler' basis functions $\{\psi_i\}_{i \in I} =: \Psi$ as $h_\theta(\cdot) := \sum_{i \in I} \theta_i \cdot \psi_i(\cdot)$. The functions $\psi_i(\cdot)$ are then often chosen as polynomials $\psi_i(\lambda) = \sum_k a_k \lambda^k$ (Defferrard et al., 2016; Kenlay et al., 2020; He et al., 2021; 2022b), so that $\psi_i(L)$ is also given as a polynomial; now in the matrix L: $\psi_i(L) = \sum_k a_k L^k$. The matrices $\{\psi_i(L)\}_{i \in I}$ are then precomputed. Complete filters $h_\theta(L)$ are parametrized via the learnable coefficients $\{\theta_i\}_{i \in I}$ as $h_\theta(L) := \sum_{i \in I} \theta_i \cdot \psi_i(L)$.

2.3 Spectral Graph Convolutional Networks:

Learnable filters are then combined into a (K-layer) graph convolutional network mapping initial node-features $X \in \mathbb{R}^{N \times F}$ to final representations $X^K \in \mathbb{R}^{N \times F_K}$. Layer-updates are implemented as

$$X^{\ell}_{i:} = \rho\left(\sum_{j=1}^{F_{\ell-1}} h^{\ell}_{\theta_{ij}}(L)(X^{\ell-1}_{j:}) + B^{\ell}_{i:}\right) \ (1) \quad \Leftrightarrow \quad X^{\ell} = \rho\left(\sum_{i\in I} \psi_i(L)\cdot X^{\ell-1}\cdot W^{\ell}_i + B^{\ell}\right) \ (2)$$

with biases $B^{\ell} \in \mathbb{R}^{N\times F_{\ell}}$ ($B_{:j} = b_j \cdot \mathbb{1}_G$) and weight matrices $W^{\ell}_i \in \mathbb{R}^{F_{\ell-1}\times F_{\ell}}$. We here consider activation functions ρ satisfying $\rho(0) = 0$ and $|\rho(a) - \rho(b)| \leqslant |a - b|$ such as e.g. (leaky-)ReLu. The scalar (1) and matrix (2) viewpoints are connected via the identity $h_{\theta_{ij}}(L) \equiv \sum_k (W_k)_{ij}\psi_k(L)$. With basis functions $\Psi = \{\psi_i\}_{i\in I}$, weights $\mathscr{W}$ and biases $\mathscr{B}$, we denote the output of a graph neural network based on the operator L and applied to the node feature matrix X as $\Phi = \Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X)$.

3 When should models be transferable? A diffusion perspective.

To determine between which graphs a GNN should be transferable, we need a measure of closeness between graphs. If graphs $G, \tilde{G}$ share a node set, an obvious first choice is the distance $\|L - \tilde{L}\|$ between their respective Laplacians. This measure is e.g. especially well adapted to the important setting of similarity under small edge variations ($w_{ij} \mapsto (w_{ij} + \delta_{ij})$ with $|\delta_{ij}| \ll 1$) (Gama et al., 2019; 2020). There do however also exist structural changes which may be considered small, but to which this standard measure $\|L - \tilde{L}\|$ is insensitive: Removing any edge from an unweighted graph G to obtain $\tilde{G}$ will always result in $2 = \|L - \tilde{L}\|$. Depending on the location of this edge removal, the graphs $G, \tilde{G}$ might however still exhibit considerable similarity: Removing a single edge in an N-clique graph K_N (Fig. 1) intuitively corresponds to a much more minor structural modification than removing the bridge-edge connecting two cliques (Fig. 2).

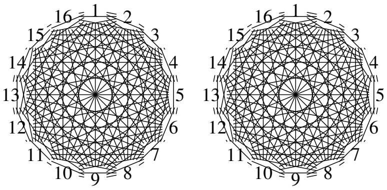

Figure 1: Left: original K_N graph Right: K_N without edge $[1 \leftrightarrow 5]$

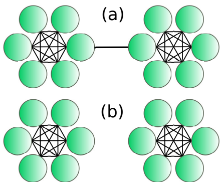

Figure 2: Dumbbell with & w/o bridge

3.1 The Notion of Diffusion Distance

This intuition that the graphs of Fig. 1 are closer to each other than those of Fig. 2 is related to the way information diffuses within them. Deleting the sole edge between cliques disrupts information flow. In contrast deleting a single edge in a high connectivity area hardly has any repercussions. To quantify this, we recall that the diffusion equation on a graph is given by $dX(t)/dt = -L \cdot X(t)$ with solution $X(t) = e^{-Lt} \cdot X(0)$. Given the same initial conditions, the maximal possible difference in diffusion-flows $X(t)$ generated by the two Laplacians $L, \tilde{L}$ at time t is

$$\eta(t) = \|e^{-Lt} - e^{-\tilde{L}t}\|.$$

In Fig. 3 we plot this difference for the graphs of Fig. 1. If $N > 2$, $\eta(t)$ only attains small values. Hence at any given time information is indeed diffused very similarly over the distinct graphs G, $\tilde{G}$.

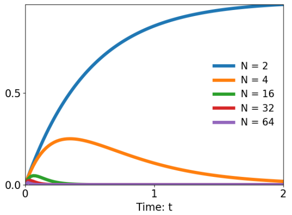

Figure 3: $\eta(t)$ for Fig. 1

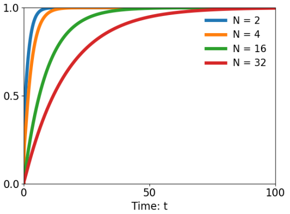

Figure 4: $\eta(t)$ for Fig. 2

Taking the supremum $\sup_{t\geqslant 0}\eta(t)$ leads to the notion of *diffusion distance* $d(G, \tilde{G}) = \sup_{t\geqslant 0}\eta(t)$ of graphs sharing a node set (Hammond et al., 2013). As N increases, this maximal overall difference becomes smaller. Hence from a diffusion perspective, K_N becomes more and more similar to its reduced version with edge removed. For K_2 instead $d(G, \tilde{G}) = 1$. Deleting the single present edge between two nodes produces a very different graph. Similarly removing the only edge that is connecting two cliques of N nodes as in Fig. 2 leads to diffusion-flow differences $\eta(t)$ that tend to one (c.f. Fig. 4). Hence the corresponding graphs are not considered similar from the perspective of diffusion. This is a sensible result, as they e.g. differ in their numbers of connected components.

Here we will hence consider two graphs to be similar if information diffuses similarly within them. For graphs sharing a node set, this is captured by the diffusion distance $d(G, \tilde{G}) = \sup_{t\geqslant 0}\eta(t)$. The exponential suppression of high-lying spectral information renders this metric adept at capturing variations preserving coarse structures (but ill-suited for fine-structure variations; c.f. Appendix C).

3.2 Generalizing Diffusion Similarity to varying numbers of Nodes

For graphs $G, \tilde{G}$ with different numbers of nodes, the diffusion processes $e^{-Lt}, e^{-\tilde{L}t}$ are defined on spaces of different dimensions. Hence they may not straightforwardly be compared. A first idea to overcome this obstacle is to consider a linear intertwining operator $J : \mathbb{R}^{|G|} \to \mathbb{R}^{|\tilde{G}|}$, transferring signals from the graph G to the graph $\tilde{G}$ (Braker Scott, 2021):

Definition 3.1. Graphs $G, \tilde{G}$ are **monodirectionally similar** under the intertwining J if $\sup_{t \geqslant 0} \|Je^{-Lt} - e^{-\tilde{L}t}J\| \ll 1$.

In this setting, we can transfer the diffusion process from G to $\tilde{G}$ without a large deviation, but generically not vice versa.

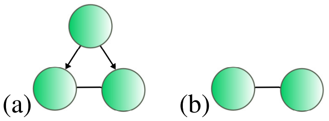

Figure 5: Monodirectionally similar graphs

Such a setting might e.g. occur if G is a subgraph of $\tilde{G}$: In the example of Fig. 5 (further discussed in Appendix G) we may transfer the diffusion process on the right hand side onto the graph on the left hand side. Transferring in the opposite direction is however impossible: Information flowing from the top node of the directed graph in Fig. 5 (a) could never be accounted for in the graph of Fig. 5 (b).

In order to establish a *reflexive* notion of similarity (where G is similar to $\tilde{G}$ and $\tilde{G}$ is also similar to G), we need to be able to transfer the diffusion process from G to $\tilde{G}$ and then also back to G again, without accruing a big error. As an example, let us consider graphs that contain clusters of nodes which are connected by significantly larger edge weights than those of edges outside of these clusters. From a diffusion perspective, information in a graph equalizes faster along edges with large weights. In the limit where edge-weights within certain sub-graphs tend to infinity, information within these clusters equalizes immediately. Such clusters thus effectively behave as single nodes. We might thus consider a coarse grained graph $\underline{G}$ where strongly connected clusters are fused together and represented only via single nodes. This naturally leads to the notion of graph coarsification, as first formalized and studied in Loukas & Vandergheynst (2018); Loukas (2019).

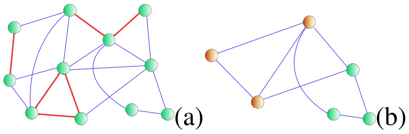

Figure 6: (a) G (stongly connected) clusters in red (b) Coarse grained $\underline{G}$

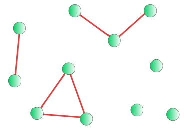

Figure 7: G_{cluster}

In our case at hand the node set $\underline{\mathcal{G}}$ of the coarse grained graph $\underline{G}$ is then given by the set of connected components in G_{cluster} (c.f. Fig 7). Edges $\underline{\mathcal{E}}$ are given by elements $(R, P) \in \underline{\mathcal{G}} \times \underline{\mathcal{G}}$ with non-zero accumulated edge weight $\underline{W}_{RP} = \sum_{r \in R} \sum_{p \in P} W_{rp}$. Node weights in $\underline{G}$ are defined accordingly by aggregating as $\underline{\mu}_R = \sum_{r \in R} \mu_r$. To compare signals on these two graphs, we define intertwining operators $J^{\downarrow}, J^{\uparrow}$ transferring information between G and $\underline{G}$: Let x be a scalar graph signal and let $\mathbb{1}_R$ be the vector that has 1 as entry for nodes $r \in R$ and is zero otherwise. Denote by u_R the entry of u at node $R \in \underline{\mathcal{G}}$. Projection $J^{\downarrow}$ is then defined component-wise by evaluation at node $R \in \underline{\mathcal{G}}$ as the average of x over R: $(J^{\downarrow}x)_R = \langle \mathbb{1}_R, x \rangle / \underline{\mu}_R$. Going in the opposite direction, interpolation is defined as $J^{\uparrow}u = \sum_{R \in \underline{\mathcal{G}}} u_R \cdot \mathbb{1}_R$. In this setting, we have (c.f. Appendix I.1) that

$$\|e^{-tL} - J^{\uparrow} e^{-t\underline{L}} J^{\downarrow}\| \lesssim 1/w_{\text{high}}^{\min} \quad \text{for any} \quad t > 0. \tag{3}$$

Here $w_{\text{high}}^{\min} \gg 1$ denotes the minimal edge weight inside the strongly connected clusters in G. As the strength of the edge-weights in G_{cluster} tends to infinity, we have by 3 that also $\eta(t) = \|e^{-Lt} - J^{\uparrow} e^{-\underline{L}t} J^{\downarrow}\| \to 0$ for any $t > 0$. Thus (for $t > 0$) the diffusion process e^{-Lt} on G acts essentially as first projecting the input-signal to $\underline{G}$ via $J^{\downarrow}$, then diffusing information on the coarse grained graph $\underline{G}$ via $e^{-\underline{L}t}$ and finally interpolating back to the original graph G via $J^{\uparrow}$. Generalizing the notion of projection and interpolation beyond coarse-graining we make the following definition:

Definition 3.2. Consider two graphs G and $\tilde{G}$ with linear intertwining operators J and $\tilde{J}$ mapping from G to $\tilde{G}$ and vice versa. We call G and $\tilde{G}$ **bidirectionally similar** if $\|e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J\| = \eta(t)$ for some (fast decaying) function $\eta(t) \geqslant 0$ with $\lim_{t \to \infty} \eta(t) = 0$ and $\eta(0) = \|Id_G - \tilde{J}J\|$.

Since G and $\tilde{G}$ typically have different numbers of nodes, we generically can not demand $\tilde{J}J = Id_G$. In the coarse graining setting above, $J (= J^{\downarrow})$ is not invertible as it maps from a larger to a smaller graph. Hence in this setting $\tilde{J}J (= J^{\uparrow}J^{\downarrow})$ will not have full rank and can thus in particular never equal the identity Id_G. We thus have $\sup_{t \geqslant 0} \eta(t) = \eta(0) = \|Id_G - \tilde{J}J\| > 0$ independent of $L, \tilde{L}$. In this bidirectional setting, similarity between the two graphs is instead measured by how fast the difference between the respective diffusion processes on G and $\tilde{G}$ becomes negligible as diffusion time t increases beyond the initial $t = 0$; i.e. by how fast $\eta(t)$ decays to zero. Exemplarily , we plot $\eta_w(t) = \|e^{-Lt} - J^{\uparrow}e^{-\tilde{L}t}J^{\downarrow}\|$ for the coarse graining setting of Figure 8: We have $\eta_w(0) \equiv \|Id_G - J^{\uparrow}J^{\downarrow}\| = 1$ irrespective of the variable edge weight w (colored red in Fig. 8). For fixed $t > 0$, we see that $\eta_w(t) \to 0$ as w increases. Additionally, the decay $\eta_w(t) \to 0$ for increasing t is faster, the larger w is chosen. This is congruent with our intuition: The stronger two nodes are connected, the more they act as a single entity.

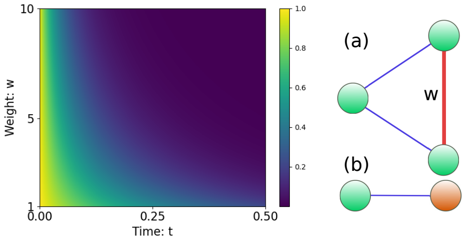

Figure 8: $\eta_w(t)$-plot for graphs (a) & (b)

4 Establishing Transferability between similar Graphs

We now characterize those filters and networks that are transferable between graphs that are similar in the mono- and bidirectional diffusion sense of Definitions 3.1 & 3.2. A discussion of the alternative setting where instead $\|L - \tilde{L}\|$ is small is provided in Appendix E. There, additional conditions on filter functions are generically necessary to guarantee transferability (Gama et al., 2019; 2020).

4.1 Laplace-Transform-Filters

In the bidirectional setting of eq. (3), this e.g. means that we want our filter function g_θ to satisfy

$$\|g_\theta(L) - J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\| \to 0 \;\text{ if } \|e^{-Lt} - J^{\uparrow}e^{-\underline{L}t}J^{\downarrow}\|_{t>0} \to 0. \tag{4}$$

In other words deploying g_θ on G should approximately result in the same outcome as first projecting to $\underline{G}$, then deploying g_θ there and finally interpolating back to G if the two graphs are similar. Typical polynomial filters ($g_\theta(L) = \theta_0 Id + \theta_1 L + \theta_2 L^2 + ...$) will not be able to satisfy (4): Here the norm of the Laplacian L on the graph G tends to infinity as at least one of the weights inside G tends to infinity ($w^{\text{min}}_{\text{high}} \to \infty$). Hence we also have $\|g_\theta(L)\| \to \infty$ for any such polynomial filter. Since on the coarse grained graph $\underline{G}$ the norm $\|g_\theta(\underline{L})\| \lneq \infty$ is constant, we have $\infty \leftarrow \|g_\theta(L)\|/2 \leqslant (\|g_\theta(L)\| - \|J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\|) \leqslant \|g_\theta(L) - J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\|$ for any polynomial g_θ. Hence the difference $\|g_\theta(L) - J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\|$ diverges and we can in particular never achieve $\|g_\theta(L) - J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\| \to 0$.

To characterize the class of filters that *can* satisfy (4), we note that as per our assumption, at any time $t > 0$ the diffusion flows over the graphs $G, \underline{G}$ are similar. Such a similarity will persist If we build up filters as a weighted sum of such diffusion flows that have progressed to various times ($g(\tilde{L}) \sim \sum_k a_k e^{-t_k\tilde{L}}$) and the coefficients $\{a_k\}_k$ are not too large. If for each time individually we have $\|e^{-Lt} - J^{\uparrow}e^{-\underline{L}t}J^{\downarrow}| < \delta$, we can estimate $\|g(\tilde{L}) - J^{\uparrow}g_\theta(\underline{L})J^{\downarrow}\| \leqslant (\sum_k |a_k|) \cdot \delta$ by a triangle-inequality argument. Making this idea precise, we hence make the following definition:

Definition 4.1. Let $\hat{\psi}$ be a (generalized) function defined on $[0, \infty)$ for which $\|\hat{\psi}\|_1 := \int_0^\infty |\hat{\psi}(t)|dt < \infty$. A **Laplace Transform Filter** (LTF) ψ is any function defined as $\psi(z) := \int_0^\infty e^{-tz}\hat{\psi}(t)dt$.

The integral in Definition 4.1 defines the *Laplace-Transform* of the (generalized) function $\hat{\psi}$ (c.f. e.g. Widder (1941) or Appendix H.2 for an introduction). The result of applying such a Laplace transform filter ψ to a characteristic operator L can then be represented as $\psi(L) = \int_0^\infty \hat{\psi}(t)e^{-tL}dt$. The term *generalized function* $\hat{\psi}$ is used in a distributional sense: We e.g. allow $\hat{\psi}(t)$ to be given as the dirac delta distribution $\hat{\psi}_{\delta_{t_0}}(t) := \delta(t - t_0)$ with $t_0 \geqslant 0$. We provide a rigorous mathematical discussion in Appendix H. Here we give two instructive examples of Laplace Transform Filters:

Example 4.2. Exponential basis functions: Considering $\hat{\psi}_k = \delta(t - kt_0)$ ($t_0 > 0$, $k \in \mathbb{N}$) yields $\psi_k(z) = e^{-(kt_0)z}$. Using this set $\Psi^{\text{Exp}} = \{e^{-(kt_0)z}\}_{k\in\mathbb{N}}$ a wide class of filter functions $h_\theta(\cdot) := \sum_i \theta_i \cdot \psi_i(\cdot)$ may be parametrized (c.f. Appendix H.2). Corresponding filters $\psi_k(L) = e^{-(kt_0)L}$ have e.g. been used in (Wang et al., 2021; 2022) to construct convolutional networks on manifolds.

Example 4.3. Resolvent basis functions: Defining $\hat{\psi}_k := (-t)^{k-1}e^{-\lambda t}$ yields $\psi_k(z) = (z + \lambda)^{-k}$. Using the set $\Psi^{\text{Res}} = \{(z + \lambda)^{-k}\}_{k\in\mathbb{N}}$ yields a function class $\{h_\theta(\cdot) := \sum_i \theta_i \cdot \psi_i(\cdot)\}$ which was theoretically investigated in Koke (2023) and is used for tasks such as node classification (Levie et al., 2019c) or molecular property prediction (Batatia et al., 2024).

4.2 Establishing Single Filter Transferability

The fact that Laplace transform filters arise as an integral over diffusion processes that have progressed to various times $t \in [0, \infty)$, indeed endows such filters with the desired transferability properties:

Theorem 4.4. As we prove in Appendix H.3, we find for the transferability of a single filter ψ that:

- $\|J\psi(L) - \psi(\tilde{L})J\| \leqslant \|\hat{\psi}\|_1 \cdot \sup_{t\geqslant 0} \|Je^{-Lt} - e^{-\tilde{L}t}J\|$ in the *monodirectional* setting.
- $\|\psi(L) - \tilde{J}\psi(\tilde{L})J\| \leqslant \int_0^\infty |\hat{\psi}(t)| \cdot \|e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J\|dt$ in the *bidirectional* setting.

In the monodirectional setting of Definition 3.1, $\|\hat{\psi}\|_1$ determines the stability constant, while the generalized diffusion distance $\sup_{t\geqslant 0} \|Je^{-Lt} - e^{-\tilde{L}t}J\|$ measures graph-similarity. Here no further restrictions on filter functions need to be imposed to guarantee (mono-directional) transferability.

In the bidirectional setting of Definition 3.2, transferability is determined by the interplay of the difference $\|e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J\| = \eta(t)$ and the (generalized) function $\hat{\psi}(t)$. As we observed in Fig. 8, we generically have $0 < \eta(0) \sim 1$ (as opposed to $\eta(0) \ll 1$), with a decay to zero for increasing t.

Hence transferability for a filter ψ is worse (i.e. the difference $\|\psi(L) - \tilde{J}\psi(\tilde{L})J\|$ is larger), the more the (finite) mass of $\hat{\psi}$ is concentrated towards the origin. In particular if $\hat{\psi}(t) = \delta(t)$, we have $\int_0^\infty |\hat{\psi}(t)|\eta(t)dt = \eta(0) = \|Id_G - \tilde{J}J\| \geqslant 0$. Thus for filters to be transferable in the bidirectional setting, the generalized function $\hat{\psi}$ may not contain any dirac-delta at $t = 0$. As we show in Appendix H.4, this is equivalent to demanding decay of the resulting filter function ψ to zero at infinity:

Corollary 4.5. *Consider a sequence of graphs G_n for which $\|e^{-L_n t} - \tilde{J}_n e^{-\tilde{L}t}J_n\||_{t>0} \to 0$. Then for a Laplace transform filter ψ, we have $\|\psi(L_n) - \tilde{J}_n\psi(\tilde{L})J_n\|| \to 0$ if and only if $\lim_{r\to\infty} \psi(r) = 0$.*

Here $J_n, \tilde{J}_n$ denote projection and interpolation operators for the n^{th} graph G_n in the sequence $\{G_n\}_n$. As a consequence of Corollary 4.5 *only* filter functions satisfying $\lim_{r\to\infty} \psi(r) = 0$ guarantee bidirectional transferability. When expanding filters as $h_\theta(L) := \sum_k \theta_k \cdot \psi_k(L)$ (c.f. Section 2.2) and using Exponential- or Resolvent basis- functions (c.f. Examples 4.2 & 4.3), this e.g. means that including the $k = 0$ term will (only) result in monodirectional transferability, while excluding it will additionally also result in bidirectional transferability.

4.3 Transferability after Filter Composition: The Network Level

We now combine filters into entire spectral convolutional networks (c.f. Section 2.3). We will assume that the basis functions $\Psi = \{\psi_i\}_{i\in I}$ utilized in equation (2) are given as Laplace Transform Filters such as the ones introduced in Examples 4.2 & 4.3. For such LTF-based architectures, we then derive transferability guarantees in terms of the learned weights & biases and – importantly – the transferability properties these basis functions $\{\psi_i\}_{i\in I}$ utilized inside the networks.

4.3.1 Node-Level Transferability

At the node level, we are interested in transferring generated node-embeddings between graphs.

Monodirectional Transferability: In this setting we start by considering initial node-features X on G. We then consider two ways of generating embeddings on the graph $\tilde{G}$: On the one hand, we may first generate node embeddings $\Phi(X)$ on G and then transfer the result to $\tilde{G}$ to obtain node embeddings $J\Phi(X)$ there. On the other hand, we may first transfer the original node-features X on G to the graph $\tilde{G}$ yielding JX. Then we may generate node-embeddings on $\tilde{G}$ using the same network Φ there, yielding $\Phi(JX)$. For the difference between these node-embeddings, we find:

Theorem 4.6. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep LTF-based network. Assume $\sum_{i\in I} \|W_i^\ell\| \leqslant W$ and $\|B^\ell\| \leqslant B$. Choose $C \geqslant \|\Psi_i(\tilde{L})\|$ $(i \in I)$ and w.l.o.g. assume $CW > 1$. Assume $\rho(JX) = J\rho(X)$. If biases are enabled, assume $J\mathbb{1}_G = \mathbb{1}_{\tilde{G}}$. Then we have with $\delta = \max_{i\in I}\{\|J\psi_i(L) - \psi_i(\tilde{L})J\|\}$:

$$\|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\tilde{L}, JX)\| \leqslant \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1}B\right)\right] \cdot \delta.$$

We prove Theorem 4.6 in Appendix H.7. We see that transferability is determined by the sizes W, B of learned weight and bias matrices, the network depth K as well as the transferability error δ of the individual basis functions. The constant C is typically of order one (e.g. in Examples 4.2 & 4.3)). Stated conditions might be relaxed (e.g. to J and ρ only almost commuting) at the cost of larger stability constants. Nevertheless, the commutativity assumption for J and ρ is e.g. satisfied for the coarse-graining example of Section 3. Similarly $J\mathbb{1}_G = \mathbb{1}_{\tilde{G}}$ is satisfied in this setting. If directed graphs are considered, it however need not be fulfilled, as we discuss further in Appendix I.3: There exist situations for which networks without biases are transferable while networks with biases are not.

Bidirectional Transferability: Here we compare node embeddings $\Phi(X)$ generated on G with node-embeddings generated by first projecting to $\tilde{G}$, applying Φ there and then translating back to G.

Theorem 4.7. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep LTF-based network. Assume that $\sum_{i\in I} \|W_i^\ell\| \leqslant W$ and $\|B^\ell\| \leqslant B$. Choose $C \geqslant \|\Psi_i(L)\|, \|\Psi_i(\tilde{L})\|$ $(i \in I)$ and w.l.o.g. assume $CW > 1$. Assume $\rho(\tilde{J}\tilde{X}) = \tilde{J}\rho(\tilde{X})$ and if biases are enabled, assume $\tilde{J}\mathbb{1}_{\tilde{G}} = \mathbb{1}_G$. Set $\max_{i\in I}\{\|\psi_i(L) - \tilde{J}\psi_i(\tilde{L})J\|\} = \delta_1$ and define $\delta_2 = \max_{i\in I}\{\|\psi_i(\tilde{L})[J\tilde{J} - Id_{\tilde{G}}]\|\}$. With this, we have that

$$\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \leqslant \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1}B\right)\right] \cdot (\delta_1 + \delta_2).$$

Here we additionally demand that $\max_{i\in I}\{\|\psi_i(\tilde{L})[J\tilde{J} - Id_{\tilde{G}}]\|\} = \delta_2$ is small to establish transferability. This is e.g. true in the coarse graining example of Section 3, where $J\tilde{J} = J^\downarrow J^\uparrow = Id_{\underline{G}}$ (as opposed to the opposite pairing $J^\downarrow J^\downarrow \neq Id_G$). In general demanding $\|\psi_i(\tilde{L})[J\tilde{J} - Id_{\tilde{G}}]\| \ll 1$ is however a much weaker condition than $[J\tilde{J} - Id_{\tilde{G}}] = 0$. We discuss this further in Appendix H.7.

4.3.2 Graph Level Transferability

Beyond node level tasks, one might also consider graph level tasks, where entire graphs are embedded into latent spaces. We first specify how graph-level latent embeddings arise:

Definition 4.8. We aggregate embeddings $X \in \mathbb{R}^{N\times F}$ of individual nodes to graph-embeddings $\Omega(X) \in \mathbb{R}^F$ as $\Omega(X)_j = \sum_{i=1}^N |X_{ij}| \cdot \mu_i$. Here $\{\mu_i\}_i$ is the set of node-weights (c.f. Section 2.1).

Given such an aggregation of node embeddings into latent-embeddings of entire graphs, we may then relegate graph-level transferability back to node-level transferability. We have (c.f. Appendix H.8)):

Theorem 4.9. Assuming $\Omega(JX) = \Omega(X)$, we have in the setting of Theorem 4.6 that
$\|\Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \leqslant \|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\|$.
Assuming $\Omega(\tilde{X}) = \Omega(\tilde{J}\tilde{X})$, we have in the (bidirectional) setting of Theorem 4.7 that
$\|\Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \leqslant \|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\|$.

The consistency assumption $\Omega(JX) = \Omega(X)$ clearly need only be satisfied on the output of the node-level network Φ; where it is e.g. satisfied for the coarse graining example of Section 3.

5 Example Settings and Validation of Theoretical Findings

Having established our theoretical results, we now showcase how they are applicable in practice.

5.1 Graph-Level Transferability between Resolutions

Let us first revisit our earlier example of graphs $G, \underline{G}$ describing the same underlying object at different resolution scales (c.f. Section 3): One original resolution-scale and one 'coarse-grained' scale, where (typically strongly connected) clusters within G are aggregated to single nodes in $\underline{G}$.

Transferability of LTF-based networks: To numerically investigate transferability of LTF-based networks in this multi-resolution setting above, we make use of the QM7 dataset (Rupp et al., 2012), consisting of graphs of organic molecules containing both hydrogen and heavy atoms. Prediction target is molecular atomization energy. Each molecule is represented by a weighted adjacency matrix, whose entries $A_{ij} = Z_i Z_j \cdot |\vec{x}_i - \vec{x}_j|^{-1}$ correspond to Coulomb repulsions between atoms i and j.

From a physical perspective, describing a molecule at the level of interacting atoms corresponds to a specific choice of resolution scale: Interactions of individual protons and neutrons inside the individual atomic nuclei are discarded. Instead only an aggregate description is used and each nucleus is described by a single node. In order to test GNN-transferability between graphs describing the same object at different resolutions, we additionally also consider a version of QM7 where we lower the resolution scale even further: Here we aggregate each heavy atomic core additionally together with its surrounding single-proton hydrogen atoms into super-nodes. Appendix J.1 provides exact details. We might interpret this QM7$_{\text{coarse}}$ dataset as a model for data obtained from a resolution-limited observation process unable to resolve positions of individual (small) hydrogen atoms and only providing information about how many hydrogen atoms are bound to a given heavy atom.

We then consider two architectures using Laplace transform filters (LTF-Exp & LTF-Res) based on the exponential and resolvent basis-functions introduced in Examples 4.2 & 4.3. We also investigate transferability properties of typical types of GNN architectures: We represent message-passing architectures through GCN, attention based methods via GATv2 and simple and advanced spectral methods via ChebNet and BernNet respectively. Pooling methods are represented through SAG. As our experiment considers graphs on different resolution scales, we also investigate transferability of methods whose propagation scheme is inherently multi-scale (SAG-M, UFGNet, Lanczos and PushNet). Using the high-resolution graphs $\{G\}$ of QM7 and the low-resolution graphs $\{\underline{G}\}$ in coarsified-QM7, we then investigate the transferability of GNNs by confronting models during inference with a resolution-scale different from the one they were trained on. Table 1 collects results.

Mean-absolute-errors (MAEs) made during inference increase significantly for methods not employing Laplace transform filters, when going from a same-resolution setting to a cross-resolution setting. Standard architectures are not transferable in the considered setting. While also such methods *can* enjoy transferability properties (Ruiz et al., 2020; Roddenberry et al., 2022; Le & Jegelka, 2023), corresponding guarantees have only been established in the setting of large graphs and thus do not apply here. As we see, also employing common multi-scale propagation schemes does not result in transferability. Cross-resolution MAEs of such methods are among the largest (of order 10^2-10^3).

Table 1: Regression using high- and low-resolution QM7

Mean Absolute Error (↓) on QM7 [kcal/mol]				
Training	**High Resolution**		**Low Resolution**	
Inference	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**
GCN	$125.34_{\pm 2.47}$	$63.17_{\pm 0.92}$	$67.75_{\pm 3.73}$	$380.51_{\pm 30.33}$
GATv2	$415.09_{\pm 96.57}$	$48.41_{\pm 19.20}$	$60.01_{\pm 3.34}$	$245.03_{\pm 90.97}$
ChebNet	$568.47_{\pm 37.70}$	$64.63_{\pm 1.21}$	$64.90_{\pm 4.55}$	$339.64_{\pm 101.30}$
SAG	$542.16_{\pm 27.33}$	$68.43_{\pm 1.93}$	$104.20_{\pm 3.92}$	$506.75_{\pm 60.57}$
BernNet	$765.22_{\pm 495.28}$	$83.76_{\pm 21.75}$	$90.52_{\pm 37.17}$	$594.62_{\pm 341.55}$
SAG-M	$285.53_{\pm 95.54}$	$66.22_{\pm 4.51}$	$73.57_{\pm 14.57}$	$307.67_{\pm 77.24}$
UFGNet	$620.21_{\pm 4.80}$	$13.71_{\pm 1.05}$	$24.53_{\pm 4.80}$	$156.44_{\pm 156.44}$
Lanczos	$939.87_{\pm 16.35}$	$10.55_{\pm 3.22}$	$83.11_{\pm 5.27}$	$654.61_{\pm 529.13}$
PushNet	$2442.59_{\pm 303.27}$	$60.94_{\pm 1.83}$	$69.25_{\pm 3.11}$	$124.08_{\pm 3.94}$
LTF-Res	$16.54_{\pm 3.01}$	$16.53_{\pm 3.03}$	$15.79_{\pm 0.98}$	$13.80_{\pm 1.34}$
LTF-Exp	$16.37_{\pm 1.71}$	$16.36_{\pm 2.16}$	$16.25_{\pm 1.41}$	$16.25_{\pm 1.41}$

MAEs of LTF-based methods do not increase when going from a same- to a cross-resolution setting: Networks based on Laplace transform filters are transferable. In cross-resolution settings, MAEs of LTF-Res and LTF-Exp are lower than that of other methods by a factor of order at least 10^1 but up to 10^2. Interestingly LTF-Res's best performance is achieved when trained on low-resolution data and deployed on high resolution test-data; a setup is likely to occur in real-life settings without high-quality training-data. We can understand these transferability results from a diffusion perspective: Numerically evaluating the left hand side of eq. (4) for graphs G in QM7 and $\underline{G}$ in QM7$_{\text{coarse}}$, we find that e.g. $\|e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\||_{t\geqslant 1} \lesssim 10^{-1}$. When investigating the differences $\|e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\| \equiv \eta(t)$ of diffusion flows, we find that $\eta(t)$ drops to zero fast, as exemplarily plotted in Fig. 9 for the first few molecules of QM7. Thus from the the perspective of diffusion, original molecular graphs G and corresponding coarse grained graphs $\underline{G}$ are close to each other. The transferability theory developed in Section 4 then explains the transferability of LTF-based networks in Table 1 (c.f. also the discussion in Appendix J.2).

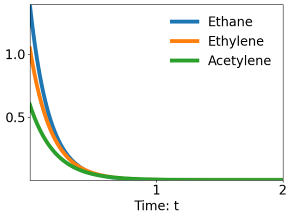

Figure 9: $\eta(t)$-plots

Continuity of LTF-based Networks: We now probe the properties of LTF-based networks even further: Theorem 4.9 guarantees that if a sequence of graphs $\{G_n\}_n$ converges to a limit graph $\underline{G}$ in the diffusion-flow sense (i.e. $\eta(t)|_{t>0}$ of Definition 3.2 approaching the constant-zero-function), the embeddings $\{F_n\}_n$ generated for the graphs $\{G_n\}_n$ will converge to the latent embedding $\underline{F}$ of $\underline{G}$.

Equation (4) now guarantees, that increasing edge-weights within the components of G_{cluster} that are being collapsed into single nodes produces graphs $\{\tilde{G}\}$ that converge (in the diffusion sense) to the coarse-grained graph $\underline{G}$. This is of course desirable: The stronger the connectivity within the connected components of G_{clsuter}, the more it is justified to treat them as the (super-)nodes making up $\underline{G}$ (c.f. Section 3). To numerically verify the convergence of corresponding latent embeddings we modify the molecular graphs of QM7 again: We now deflect hydrogen atoms (H) out of their equilibrium positions towards the respective nearest heavy atom. This then introduces a setting precisely as discussed: Edge-weights $A_{ij} = Z_i Z_j \cdot |\vec{x}_i - \vec{x}_j|^{-1}$ between heavy atoms remain the same, while those between H-atoms and nearest heavy atomic nuclei increasingly diverge. We then compare embeddings $\{\underline{F}\}$ generated for coarsified graphs $\{\underline{G}\}$, with embeddings $\{\tilde{F}\}$ of graphs $\{\tilde{G}\}$ where hydrogen atoms have been deflected. As is evident from Figure 10, the transferability error of LTF-Res and LTF-Exp converges towards zero. We might thus think of LTF-based models as continuously mapping from the space of graphs (equipped with the diffusion-flow topology) to the Euclidean latent space. For other models, the latent distance $\|\tilde{F} - \underline{F}\|$ does not tend to zero. Thus these models can not be considered continuous. As we explore further in Appendix K, the underlying reason is that as $\tilde{G} \to \underline{G}$ in the diffusion-flow sense, information propagation inside such models is more and more governed by an effective propagation graph which is decidedly different from $\underline{G}$.

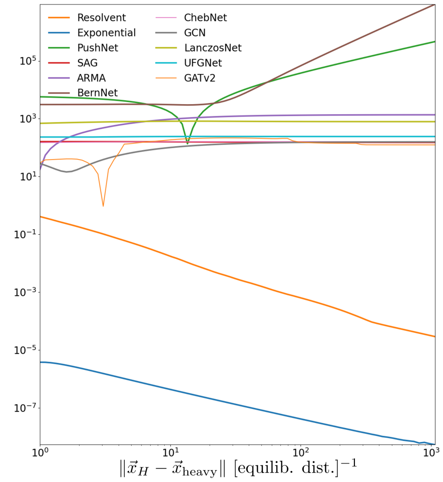

Figure 10: Latent distance $\|\tilde{F} - \underline{F}\|$

5.2 Node Level Transferability and Graphs with varying Connectivity

We next consider popular citation networks (c.f. Appendix J.5 where each node corresponds to a piece of scientific writing. Labels correspond to the academic discipline of the paper and an edge implies a citation. We then expand individual nodes into connected k-cliques (c.f. Fig. 11). We might interpret this as further dissecting each article into subsections, which reference each other.

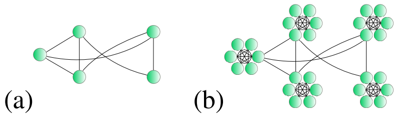

Figure 11: Individual nodes (a) replaced by k-cliques (b)

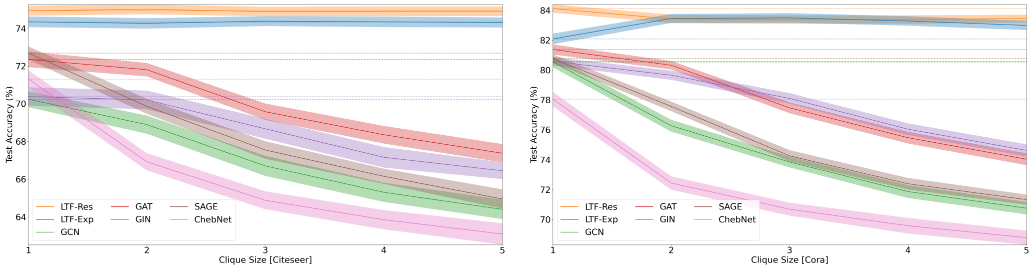

Figure 12: Node-Classification-Accuracy (↑) and uncertainty (for 100 runs) vs. clique size.

Both typical models (c.f. Appendix J.5) and LTF-based methods were then trained on the same (k-fold expanded) train-set and asked to classify nodes in the (k-fold expanded) test-partition. The classification accuracy of methods not employing Laplace Transform filters decreases significantly with increasing clique size (c.f. Fig. 12). We can understand the underlying reason for this using GCN as an Example (c.f. Appendix K for other methods): Inside a GCN-layer, a node feature matrix X is updated as $X \mapsto \hat{A}XW$, with the renormalized adjacency matrix $\hat{A}$ given as $\hat{A}_{ij} \sim A_{ij}/\sqrt{d_i d_j}$. As the degree d_i of each node increases (linearly) with increasing clique-size k, the message-strength $\hat{A}_{ij}$ between the respective cliques decreases as $\hat{A}_{ij} \sim 1/k$. Hence information propagation between the cliques becomes disrupted as k increases: GCN is more and more transferable between the given graph and a modified version where edges *between* cliques are removed. This is not the case for LTF-based networks since *from a diffusion perspective*, original- and disconnected graphs are *not* similar (c.f. Fig. 4). Hence such models are able to propagate information also *between* high connectivity areas and thus are able to retain a high classification accuracy.

5.3 Transferability between Graphs discretizing a common Manifold

The concept of operators capturing the geometry of underlying spaces also applies to manifolds $\mathcal{M}$, where the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ can be thought of as a continuous analogue of the Graph Laplacian (Hein et al., 2006). This is hence is a prime setting for studying transferability. Counter to previous works (Levie et al., 2019a; Wang et al., 2021), our diffusion framework here allows to derive transferability guarantees beyond the settings of bandlimited signals and probabalistic guarantees: We consider the setting of two graphs G_1, G_2 discretely approximating the same manifold (c.f. e.g. Fig. 13). This can be made mathematically precise using the concept of generalized norm resolvent convergence (c.f. e.g. (Post, 2012) for a discussion). Here we note the following: Given projection operators $J_i^{\downarrow}$ mapping from $\mathcal{M}$ to G_i and interpolation operators $J_i^{\uparrow}$ mapping from G_i to $\mathcal{M}$, we may measure the difference $\|e^{-t\Delta_{\mathcal{M}}} - J_i^{\uparrow} e^{-tL_i} J_i^{\downarrow}\| \leqslant \delta_i$ in diffusion flows on the respective spaces. The fidelity of the discrete approximation is then essentially determined by the size of $\delta_i \ll 1$. As discussed in detail in Appendix I.2, we have in this setting:

$$\|e^{-tL_1} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-tL_2} (J_2^{\downarrow} J_1^{\uparrow})\| \lesssim (\delta_1 + \delta_2) \qquad (5)$$

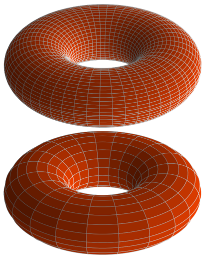

Figure 13: Torus Discretizations

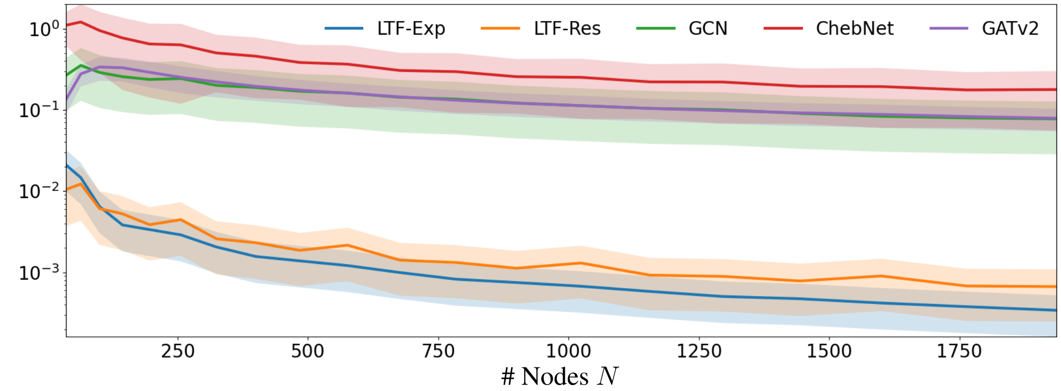

Figure 14: Transferability error $E = \|\Phi_1(J_1^{\downarrow} f) - (J_1^{\downarrow} J_2^{\uparrow})\Phi_2(J_2^{\downarrow} f)\|$ vs. # Nodes $N = |G_2| = 4|G_1|$

If $\delta_1, \delta_2 \ll 1$, the graphs G_1 and G_2 are thus bidirectionally similar in the sense of Definition 3.2. As an Example, we prove in Appendix I.2 that for the regular grid discretisation of the Torus and judiciously chosen translation operators $J_i^{\uparrow} J_i^{\downarrow}$, we have $\|e^{-t\Delta_{\mathcal{M}}} - J_i^{\uparrow} e^{-tL_i} J_i^{\downarrow}\||_{t>0} \leqslant \delta_i \to 0$ as the number of nodes in the approximating graphs G_i is increased. Given a fixed input signal $f \in L^2(\mathcal{M})$ on the Torus $\mathcal{M}$, eq. (5) together with Theorem 4.6 then implies that thus also the transferability error $E = \|\Phi_1(J_1^{\downarrow} f) - (J_1^{\downarrow} J_2^{\uparrow})\Phi_2(J_2^{\downarrow} f)\|$ tends to zero as N increases. This error E measures the difference between sampling the signal f on $\mathcal{M}$ to G_1 and passing it through a GNN there, versus sampling f to G_2, applying the GNN on G_2 instead and subsequently transfering the output to G_1. To numerically verify, that this transferability error indeed tends to zero for LTF-based methods, we fix the number of nodes as $N = |G_2| = 4|G_1|$ in the respective graphs. We then plot E as a function of the number of nodes N for randomly initialized networks, with uncertainty calculated over 100 initializations. Appendix J.6 contains additional details. As evident from Fig. 13, the transferability error for LTF-based methods tends to zero as N is increased. Additionally transferability errors of LTF-based methods are consistently two orders of magnitude smaller than those of other networks.

6 Conclusion

We developed a novel approach to transferability based on the intrinsic notion of diffusion on graphs, which considers graphs to be similar if their rough overall structures align. Transferability of entire networks in this setting was relegated to the filter functions employed inside their convolutional blocks. A rigorous analysis established that when the rough overall information whithin graphs is paramount, networks are transferable if filters arise as Laplace transforms while other filter choices will not lead to transferability. In example settings – including settings not covered by other already established approaches to transferability – this was then confirmed numerically.

REFERENCES

Wolfgang Arendt. APPROXIMATION OF DEGENERATE SEMIGROUPS. *Taiwanese Journal of Mathematics*, 5(2):279 – 295, 2001. doi: 10.11650/twjm/1500407337. URL `https://doi.org/10.11650/twjm/1500407337`.

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=-qh0M9XWxnv`.

Oscar F. Bandtlow. Estimates for norms of resolvents and an application to the perturbation of spectra. *Mathematische Nachrichten*, 267(1):3–11, 2004. doi: https://doi.org/10.1002/mana.200310149. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/mana.200310149`.

Ilyes Batatia, Lars Leon Schaaf, Gabor Csanyi, Christoph Ortner, and Felix Andreas Faber. Equivariant matrix function neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=yrgQdA5NkI`.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Francesco Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3496–3507, 2019.

L. C. Blum and J.-L. Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=0pdSt3oyJa1`.

C. Braker Scott. *Diffusion Distance: Efficient Computation and Applications*. PhD Thesis. UNIVERSITY OF CALIFORNIA IRVINE, 2021.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=F72ximsx7C1`.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

Julian Busch, Jiaxing Pi, and Thomas Seidl. Pushnet: Efficient and adaptive neural message passing. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang (eds.), *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 1039–1046. IOS Press, 2020. doi: 10.3233/FAIA200199. URL `https://doi.org/10.3233/FAIA200199`.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. ISSN 00361429. URL `http://www.jstor.org/stable/2949580`.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Diffusion scattering transforms on graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BygqBiRcFQ.

Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. *IEEE Trans. Signal Process.*, 68:5680–5695, 2020. doi: 10.1109/TSP.2020.3026980. URL https://doi.org/10.1109/TSP.2020.3026980.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

David K. Hammond, Yaniv Gur, and Chris R. Johnson. Graph diffusion distance: A difference measure for weighted graphs based on the graph laplacian exponential kernel. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 419–422, 2013. doi: 10.1109/GlobalSIP.2013.6736904.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 14239–14251, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/76f1cfd7754a6e4fc3281bcccb3d0902-Abstract.html.

Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. In *NeurIPS*, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/2f9b3ee2bcea04b327c09d7e3145bd1e-Abstract-Conference.html.

Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited, 2022b.

Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 8:1325–1368, 2006. URL https://api.semanticscholar.org/CorpusID:1355782.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90021-7. URL https://www.sciencedirect.com/science/article/pii/0378873383900217.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Tosio Kato. *Perturbation theory for linear operators; 2nd ed.* Grundlehren der mathematischen Wissenschaften : a series of comprehensive studies in mathematics. Springer, Berlin, 1976. URL https://cds.cern.ch/record/101545.

Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of polynomial spectral graph filters. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5350–5354, 2020. doi: 10.1109/ICASSP40776.2020.9054072.

Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of graph convolutional neural networks under edge rewiring. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pp. 8513–8517. IEEE, 2021a. doi: 10.1109/ICASSP39728.2021.9413474. URL https://doi.org/10.1109/ICASSP39728.2021.9413474.

Henry Kenlay, Dorina Thanou, and Xiaowen Dong. Interpretable stability bounds for spectral graph filters. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5388–5397. PMLR, 2021b. URL http://proceedings.mlr.press/v139/kenlay21a.html.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Christian Koke. Limitless stability for graph convolutional networks. In *11th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XqcQhVUr2h0.

Christian Koke. Strong connectivity in graphs: Norm resolvent convergence to effective descriptions, 2024.

Christian Koke and Daniel Cremers. Holonets: Spectral convolutions do extend to directed graphs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EhmEwfavOW.

Thien Le and Stefanie Jegelka. Limits, approximation and size transferability for GNNs on sparse graphs via graphops. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=kDQwossJuI.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3734–3743. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19c.html.

Ron Levie, Michael M. Bronstein, and Gitta Kutyniok. Transferability of spectral graph convolutional neural networks. *CoRR*, abs/1907.12972, 2019a. URL http://arxiv.org/abs/1907.12972.

Ron Levie, Elvin Isufi, and Gitta Kutyniok. On the transferability of spectral graph filters. *CoRR*, abs/1901.10524, 2019b. URL http://arxiv.org/abs/1901.10524.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Trans. Signal Process.*, 67(1): 97–109, 2019c. doi: 10.1109/TSP.2018.2879624. URL https://doi.org/10.1109/TSP.2018.2879624.

Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S. Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BkedznAqKQ.

Andreas Loukas. Graph reduction with spectral and cut guarantees. *J. Mach. Learn. Res.*, 20: 116:1–116:42, 2019. URL https://jmlr.org/papers/v20/18-680.html.

Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3243–3252. PMLR, 2018. URL http://proceedings.mlr.press/v80/loukas18a.html.

Sohir Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: an extended graphon approach. *CoRR*, abs/2109.10096, 2021. URL https://arxiv.org/abs/2109.10096.

Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Inf. Retr.*, 3(2):127–163, 2000. doi: 10.1023/A:1009953814988. URL https://doi.org/10.1023/A:1009953814988.

Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *CoRR*, abs/1905.09550, 2019. URL http://arxiv.org/abs/1905.09550.

Olaf. Post. *Spectral Analysis on Graph-like Spaces / by Olaf Post.* Lecture Notes in Mathematics, 2039. Springer Berlin Heidelberg, Berlin, Heidelberg, 1st ed. 2012. edition, 2012. ISBN 3-642-23840-8.

T. Mitchell Roddenberry, Fernando Gama, Richard G. Baraniuk, and Santiago Segarra. On local distributions in graph signal processing. *IEEE Trans. Signal Process.*, 70:5564–5577, 2022. doi: 10.1109/TSP.2022.3223217. URL https://doi.org/10.1109/TSP.2022.3223217.

Luana Ruiz, Luiz F. O. Chamon, and Alejandro Ribeiro. Graphon neural networks and the transferability of graph neural networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/12bcd658ef0a540cabc36cdf2b1046fd-Abstract.html.

M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108:058301, 2012.

Siddhartha Sahi. Harmonic vectors and matrix tree theorems, 2013.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2157.

T. Tao. *An Introduction to Measure Theory*. Graduate studies in mathematics. American Mathematical Society, 2013. ISBN 9781470409227. URL https://books.google.de/books?id=SPGJjwEACAAJ.

Gerald Teschl. *Mathematical Methods in Quantum Mechanics*. American Mathematical Society, 2014.

J. J. P. Veerman and Robert Lyons. A primer on laplacian dynamics in directed graphs. *Nonlinear Phenomena in Complex Systems*, 2020. URL https://api.semanticscholar.org/CorpusID:211066395.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:248987544.

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Stability of neural networks on riemannian manifolds. *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1845–1849, 2021. URL https://api.semanticscholar.org/CorpusID:232110514.

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Convolutional neural networks on manifolds: From graphs and back. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pp. 356–360, 2022. doi: 10.1109/IEEECONF56349.2022.10051964.

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Stability to deformations of manifold filters and manifold neural networks. *IEEE Trans. Signal Process.*, 72:2130–2146, 2024a. doi: 10.1109/TSP.2024.3378379. URL https://doi.org/10.1109/TSP.2024.3378379.

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Geometric graph filters and neural networks: Limit properties and discriminability trade-offs. *IEEE Trans. Signal Process.*, 72:2244–2259, 2024b. doi: 10.1109/TSP.2024.3392360. URL https://doi.org/10.1109/TSP.2024.3392360.

David Vernon Widder. *The Laplace Transform*, volume vol. 6 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1941.

T.P. Wihler. On the hölder continuity of matrix functions for normal matrices. *Journal of inequalities in pure and applied mathematics*, 10(4), Dec 2009. ISSN 1443-5756. URL https://www.emis.de/journals/JIPAM/images/276_09_JIPAM/276_09_www.pdf.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Xuebin Zheng, Bingxin Zhou, Junbin Gao, Yuguang Wang, Pietro Lió, Ming Li, and Guido Montúfar. How framelets enhance graph neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12761–12771. PMLR, 2021. URL http://proceedings.mlr.press/v139/zheng21c.html.

Dongmian Zou and Gilad Lerman. Graph convolutional neural networks via scattering. *Applied and Computational Harmonic Analysis*, 49(3):1046–1074, nov 2020. doi: 10.1016/j.acha.2019.06.003. URL https://doi.org/10.1016%2Fj.acha.2019.06.003.

A NOTATION

We provide a summary of employed notational conventions:

Table 2: Notational Conventions

Symbol	Meaning
G	a graph
$\mathcal{G}$	Nodes of the graph G
$\mathcal{E}$	Edges of the graph G
N	number of nodes $\lvert\mathcal{G}\rvert$ in G
$\underline{G}$	Coarse grained version of graph G
μ_i	weight of node i
M	weight matrix
$\langle \cdot, \cdot \rangle$	inner product
A	(weighted) adjacency matrix
$D^{\text{in/out}}$	in/out-degree matrix
L^{in}	in-degree graph Laplacian
L, Δ	Graph Laplacian
$\Delta_{\mathcal{M}}$	Manifold Laplacian / Laplace Beltrami operator
$\nu(L)$	departure from normality of L
$\sigma(L)$	spectrum (i.e. collection of eigenvalues) of L
h	a filter function
$h(L)$	function h applied to operator L
Ψ	a filter bank
ψ_i	an element of a filter-bank
$J^{\downarrow}, J^{\uparrow}$	projection and interpolation operator
$J, \tilde{J}$	intertwining operators
Φ	map associated to a graph convolution network
Ω	graph-level aggregation mechanism
Z_i	atomic charge of atom corresponding to node i
$\vec{x}_i$	Cartesian position of atom corresponding to node i
$\frac{Z_i Z_j}{\lvert\vec{x}_i - \vec{x}_j\rvert}$	Coulomb interaction between atoms i and j
$\lvert\vec{x}_i - \vec{x}_j\rvert$	Euclidean distance between x_i and x_j

B FURTHER DISCUSSION OF EXISTING APPROACHES TO TRANSFERABILITY

In this section we provide further details on existing approaches to transferability of graph neural networks:

Graphon Neural Networks and the Transferability of Graph Neural Networks (Ruiz et al., 2020): This seminal work explores the theoretical underpinnings of Graph Neural Networks (GNNs) in the context of graphons, a mathematical generalization of graphs to large-scale, continuous structures. The paper establishes a connection between GNNs and graphons, providing insights into the behavior of GNNs on large, dense graphs ($\lvert\mathcal{E}\rvert$ is of $\mathcal{O}(N^2)$, with N the number of nodes (Le & Jegelka, 2023)) by modeling these graphs as graphons. This framework helps understand how GNNs operate in the limit of large graphs and their potential to generalize across different graph structures in this realm. A central focus of the paper is the transferability of GNNs—specifically, their ability to perform well on large graphs that may differ in size or topology from those seen during training. Transferability errors between graphs discretizing the same graphon are established to be of $\mathcal{O}(N^{-\frac{1}{2}})$, with N the minimum number of nodes. Assumptions on considered filter functions are that they are bounded and Lipschitz continuous (c.f. AS2 on page 6; ibid.).

Transferability of Graph Neural Networks: an Extended Graphon Approach (Maskey et al., 2021): This work is in spirit similar to (Ruiz et al., 2020) whose results it extends from considering the adjacency matrix as the graph shift operator to more general graph shift operators and from considering only polynomial filters to allowing for general continuous filter functions.

Limits, approximation and size transferability for GNNs on sparse graphs via graphops (Le & Jegelka, 2023): In contrast to approaches using graphons, which focus on large *dense* graphs, this paper instead focuses on transferability on *sparse* graphs ($|\mathcal{E}| = \mathcal{O}(N)$). The paper makes use of the concept of Graphops, a mathematical operator that can be used to model how GNNs behave on large sparse graphs. This operator helps analyze the limit behavior of GNNs, capturing the way information is propagated through large sparse graph structures.

One of the focuses of the paper is size transferability, which refers to the ability of a GNN to generalize across graphs of different sizes. The authors explore how GNNs can transfer learned representations from smaller, sparse graphs to larger ones, and vice versa. By leveraging the Graphop framework, the paper formalizes conditions for successful transferability between graphs of varying sizes.

On Local Distributions in Graph Signal Processing (Roddenberry et al., 2022): Thiw work is rooted in the field of graph signal procesing (GSP) and puts a particular emphasis on the transferability of GSP techniques across different graph structures. The paper focuses on the concept of graphings, which are a probabilistic framework for representing large sparse graphs and their underlying structures.

The paper investigates how local signal behaviors, defined by local distributions over neighborhoods in a graph, can be transferred from one graph to another. Specifically, it formalizes how GSP techniques—such as filtering and node classification—can be transferred to graphs that are not identical but share similar local structures.

By modeling large graphs through graphings, the authors provide a framework that makes it possible to generalize local distributions and signal processing tasks across different graphs.

Graph Convolutional Neural Networks via Scattering (Zou & Lerman, 2020) This work provides a different perspective on Graph Convolutional Networks (GCNs) by connecting them to scattering transforms, a concept from signal processing. The authors demonstrate that GCNs can be interpreted as a discrete graph counterpart of scattering transforms, which involve multi-scale wavelet-like operations that capture hierarchical information across different levels of graph structure. This connection highlights the multi-scale nature of GCNs, similar to scattering transforms, which analyze signals at varying resolutions.

A key focus of the paper is the stability of GCNs when viewed through the scattering framework. The authors argue that scattering transforms offer a more stable approach to graph signal processing compared to traditional GCNs, especially in the presence of noisy or incomplete graph data. The multi-layer structure of GCNs, when interpreted as a series of scattering operations, allows for more robust signal propagation across the graph, making GCNs less sensitive to perturbations in the graph topology.

By linking GCNs with scattering transforms, the paper provides both a theoretical foundation for understanding GCNs' operations and an approach to improving their robustness and interpretability in graph-based learning tasks.

Derived single filter transferability results depend on spectral properties of the utilized Laplacians on the respective graphs. The conditions on the spectrum also arise from a Lipschitz type approach to bounding differences, where the difference $\|\psi(L) - \psi(\tilde{L})\|$ is then via a triangle inequality argument reduced to bounding each term $\|\psi(\lambda_k)u_k u_k^\intercal - \psi(\tilde{\lambda}_k)\tilde{u}_k\tilde{u}_k^\intercal\|$ individually. This is done in eq.s (64) and (68) respectively, which are condingent the there stated spectral restrictions.

Limitless transferability for graph convolutional Networks (Koke, 2023): This work studies stability- and transferability proeprties of spectral graph neural networks, with a particular focus on directed graphs. In spirit, it is the closest to our work here, as one of the main class of filters it investigates is the class of resolvent based filter functions which constitute an example (i.e. Example 4.3) of the more general class of Laplace transform filters considered in this present work.

Stability to Deformations of Manifold Filters and Manifold Neural Networks (Wang et al., 2024a) :

This work explores the theoretical foundation of manifold filters and manifold neural networks (MNNs), focusing on their transferability across manifolds. Similarly to the filters analyzed in the present work, manifold filters are defined in terms of Laplace transforms. By framing graph neural networks (GNNs) as discrete approximations of MNNs, the authors analyze conditions under which MNNs remain stable under smooth deformations of the manifold.

Stability is shown to depend on specific spectral properties of the filter functions, including Lipschitz continuity and integral Lipschitz continuity, which control the trade-off between robustness and frequency discriminability. The paper establishes that filters meeting these conditions can generalize effectively to new manifolds by adapting to changes in the Laplace-Beltrami operator's spectrum.

More techicalle, filters are bounded as $|\psi(L) - \psi(\tilde{L})| \leqslant K\|L - \tilde{L}\|$. In Theorem 2 absolute perturbations are considered ($\tilde{L} = L + A$), in Theorem 3 relative perturbations are considered ($\tilde{L} = L + EL$). In both cases the conditions on spectrum and filter functions stem from the fact that Lipschitz-ness does not directly translate to operator Lipschitz-ness when measured in spectral norm (see e.g. Wihler (2009) for a discussion).

Geometric Graph Filters and Neural Networks: Limit Properties and Discriminability Trade-offs (Wang et al., 2024b): Here instead of measuring the linear norm difference $\|LP - \mathcal{L}P\|$ between a graph Laplacian L and a manifold Laplacian $\mathcal{L}$ (which generically would be infinite as $\mathcal{L}$ is an unbounded operator), the difference of the action of these operators on eigenfunctions ($\|LP\phi - \mathcal{L}P\phi\|$). After a triangle inequality argument, one term that has to be bounded in order to bound the difference in filter outputs is $\|\phi_i^n - \phi_i\|$ of the i^{th} eigenfunction and eigenvector respectively. The fidelity of this approximation depends on spectral separation properties (c.f. Theorem 4 ibid.), which hence leads to the requirement that the spectrum be α-separated. This requirement can thus be considered an artifact of considering the linear approximation $\|\phi_i^n - \phi_i\|$ for each eigenfunction. In contrast, in our approach (c.f. Appendix F.2) the notion of approximation of the Laplacian on the underlying manifold is different. We bound the quantity $\|J^{\uparrow}e^{-tL}J^{\downarrow} - e^{-t\Delta}\|$ instead. Hence we do not need to bound differences between individual eigenfunctions and eigenvectors and hence avoid dependencies on spectral separations.

Transferability of Spectral Graph Convolutional Neural Networks (Levie et al., 2019a): As one of the earliest works challenging the then prevailing belief that spectral methods are not transferable, this work was among the first to present theoretical proofs and experimental evidence to demonstrate that these methods can generalize effectively under certain conditions.

The key contribution is a theoretical framework in which transferability depends on how well graphs approximate a shared underlying continuous domain, such as a topological space or metric-measure space. Many graph convolutional networks are then shown to have "principle transferability" in this setting, meaning that their ability to generalize is built-in and does not rely on additional training. The analysis introduces the transferability inequality, which bounds the generalization error of filters based on the graph Laplacian's approximation quality and sampling consistency.

The study also develops sufficient conditions for achieving low transferability errors, demonstrating that spectral ConvNets can perform consistently across graphs with varying sizes, topologies, and dimensions, provided the graphs discretize the same continuous domain.

As in our work, filters here are only required to be bounded and Lipschitz continuous (c.f. Theorem 17 ibid.). However, signals are assumed to be bandlimted. We avoid Levie's growth of the stability constant with the number of considered eigenvalues (c.f. the discussion towards the end of page 12 ibid.) by avoiding approximations of individual eigenfunctions and instead approximating the bounded operator $e^{-t\Delta}$ directly.

Diffusion Scattering Transforms on Graphs (Gama et al., 2019): This work emphasizes the stability of scattering-based representations against perturbations in graph topology and reindexing. By extending the concept of scattering transforms to graph-structured data, the framework introduces diffusion scattering transforms that leverage diffusion operators to capture multi-scale hierarchical features of graph signals.

The authors focus on ensuring that the transforms are robust to changes in graph structure, such as modifications to edge weights or topology. Stability is achieved through the use of diffusion wavelets, which provide a principled way to construct graph filters that are invariant to local perturbations while retaining sensitivity to meaningful global graph features. The stability analysis demonstrates that the scattering transform bounds the impact of graph perturbations in terms of the changes they induce in the graph Laplacian's spectrum, ensuring reliable performance across varied graph inputs.

Here the dependence in Theorem 5.3 on the 'spectral gap' as defined before Proposition 4.1 comes from the Lipschitz type argument used in eq. (48).

Stability Properties of Graph Neural Networks (Gama et al., 2020): This paper investigates the stability properties of Graph Neural Networks (GNNs) to perturbations in the underlying graph structure. The authors analyze how small changes in graph topology— such as modifications to edge weights, addition or deletion of edges, or reindexing of nodes—affect the outputs of GNNs.

The paper develops a rigorous mathematical framework to assess the stability of GNNs using tools from spectral graph theory. It establishes that GNNs are stable to localized perturbations in the graph topology, with the degree of stability depending on the spectral properties of the graph filters used within the network. Specifically, it is shown that GNNs exhibit a trade-off between stability and discriminability: filters that are more stable to perturbations may sacrifice sensitivity to high-frequency information, which can limit their ability to differentiate fine-grained graph structures.

Here as well, Lipschitz type arguments are being used (See e.g. the assumptions of Theorem 1) to establish single filter transferability. Since scalar Lipschitzness does not translate to operator Lipschitzness under spectral norm, additional restrictions on spectrum and filter functions need to be hence imposed.

Following this, the authors highlight the importance of filter design in achieving a balance between robustness and expressivity. Filters that adhere to conditions such as Lipschitz continuity or integral Lipschitz continuity are particularly effective in maintaining stability while preserving key graph features.

C Comparison of diffusion similarity with standard Norm-similarity

In contrast to previous works, we do not use the norm difference $||L - \tilde{L}||$ to measure graph similarity. Instead, the distance measure we are considering is the diffusion distance

$$d(L, \tilde{L}) = \sup_{t \geqslant 0} ||e^{-tL} - e^{-t\tilde{L}}||,$$

introduced by Hammond et al. (2013).

From a spectral perspective, the key idea here is that including the exponential into the distance metric leads to an (exponential) suppression of large eigenvalues of L and $\tilde{L}$. Information encoded into these large eigenvalues (and corresponding eigenspaces) corresponds to fine structure details of the graphs G and $\tilde{G}$ (c.f. e.g. Chung (1997)).

Suppressing this fine-structure information before taking a distance measurement effectively leads to a comparison that is predominantly determined by the coarse structures within the graphs. If the rough structures within the two graphs are similar, the distance between the two graphs will then be relatively small. Thus this metric is adapted to considering graphs that are similar up to fine-structure variations to be close to each other. This is the setting we are interested in when considering transferability, so that this distance measure is adapted to this setting of transferring filters between approximately similar graphs (see also the discussion in Section 3).

In the original pape that first introduced this notion of graph similarity (Hammond et al., 2013), the authors showed diffusion distances ($d(\cdot, \cdot)$) to be a well defined metric on the space of graphs. Here, 'metric' is used in the strictly mathematical sense (i.e. satisfying the defining properties of positivity, symmetry and the triangle inequality). Hence the notion of diffusion similarity equips the space of graphs with a well defined (metric-)topology. This topology respects the one induced by Euclidean norms: If $\|L_n - L\| \to 0$ for one (and hence all) Euclidean norm, then also $d(L_n, L) \to 0$.

At the same time, the metric $d(\cdot,\cdot)$ arising from diffusion similarity is able to capture more general settings of graph similarity: One example is a sequence of graphs where the connectivity in certain subgraphs increases (c.f. Section 3.2). Such a sequence does not converge in any Euclidean norm. However, in the diffusion-distance metric it is Cauchy and hence also convergent. The limit is a coarse grained graph, where strongly connected clusters are collapsed to single nodes. Thus this diffusion based metric is e.g. naturally able to capture convergence to graphs of reduced size.

Additionally, the notion of diffusion similarity is not limited to the setting of coarse-graining graphs. Other examples settings captured by this notion of diffusion similarity are rewiring operations in graphs, the inclusion of subgraphs, or graphs discretizing the same ambient space. Additionally the notion of diffusion similarity naturally extends to directed graphs.

Hence it is indeed fair to conclude that diffusion similarity is a well-adapted and widely applicable notion of graph similarity.

D Discussion of ’Restricted Spectral Similarity’ (Loukas, 2019) and implications in the graph coarsening setting

A well established notion of graph similarity is that of ’Restricted Spectral Similarity’ (Loukas, 2019).

This notion is adapted to approximations of properties of a graph through a reduced version while preserving its fundamental spectral characteristics within a restricted subspace. This measure extends the concept of spectral similarity, commonly used in graph sparsification, to scenarios where the reduced graph has fewer vertices than the original, thus operating on a lower-dimensional space.

Spectrally restricted similarity ensures that the eigenvalues and eigenspaces of the reduced graph closely align with those of the original graph for a specified subset of eigenmodes. This property guarantees that critical features, such as cuts and the behavior of algorithms reliant on spectral embeddings (e.g., spectral clustering), remain well-approximated in the reduced graph. Theoretical results demonstrate that preserving this restricted spectral similarity leads to robust graph reduction techniques that maintain essential graph properties and enable the effective use of the reduced graph for tasks like unsupervised learning or partitioning.

In the context of the setting in our paper, restricted spectral similarity is *almost* able to guarantee transferability between an original graph and its coarse grained version:

Consider two graphs L and L_c. Using the notation of ’Andreas Loukas, Graph reduction with spectral and cut guarantees’, we are interested in bounding the difference in filter outputs $\|g(L) - \quad P^{\intercal}g(L_c)P\|$. Let us exemplarily consider the case $g(z) = e^{-z}$ (corresponding to $\hat{g}(t) = \delta(t-1)$).

Denote by Q, Q_c the spectral projections onto the first k eigenvectors of L, L_c respectively. Denote by $\tilde{Q}, \tilde{Q}_c$ the respective spectral projections onto the remaining eigenvectors of the respective two operators.

We may first observe that we may reduce the problem to considering only the first k eigenvectors of the respective operators:

$$\begin{aligned}
\|g(L) - \quad P^{\intercal}g(L_c)P\| &= \|e^{-L} - \quad P^{\intercal}e^{-L_c}P\| \\
&= \|Qe^{-L}Q - \quad P^{\intercal}Q_c e^{-L_c}Q_c P\| + \|\tilde{Q}e^{-L}\tilde{Q} - \tilde{Q}_c \;\; P^{\intercal}e^{-L_c}\tilde{Q}_c P\| \\
&\leqslant \|Qe^{-L}Q - \quad P^{\intercal}Q_c e^{-L_c}Q_c P\| + \max\{e^{-\lambda_{(k+1)}}, e^{-\lambda_{c,(k+1)}}\} \\
&= \; \|Qe^{-L}Q - \quad P^{\intercal}Q_c e^{-L_c}Q_c P\| + \mathcal{O}(\epsilon).
\end{aligned}$$

We may decompose $Q\; e^{-L}Q$ into a sum over one dimensional eigenspaces as

$$Qe^{-L}\;Q = \sum_{i=1}^{k} e^{-\lambda_i} v_i \langle v_i, \cdot\rangle$$

with eigenvectors $\{v_k\}_k$.

Similar considerations also hold for the coarse grained graph. Using this, we find

$$\begin{aligned}&\|Qe^{-L}Q - \quad P^{\intercal}Q_c e^{-L_c}Q_c P\| \\ \leqslant &\| \sum_{i=1}^{k}(e^{-\lambda_i} \ - e^{-\lambda_{c,i}} \quad)v_i \left\langle v_i, \cdot \right\rangle \quad \| + \| \sum_{i=1}^{k}(e^{-\lambda_i}(v_i \left\langle v_i, \cdot \right\rangle \ - \ P^{\intercal}v_{c,i} \left\langle \ v_{c,i} \ , P\cdot \right\rangle) \ \ \|\end{aligned}$$

The first term is then bounded by a small quantity, as Theorem 13 of 'Andreas Loukas, Graph reduction with spectral and cut guarantees' guarantees that $\lambda_i \approx \lambda_{c,i}$ for $i \leqslant k$.

For the second term we note that we may bound

$$\| \sum_{i=1}^{k}(e^{-\lambda_i}(v_i \left\langle v_i, \cdot \right\rangle \ - \ P^{\intercal}v_{c,i} \left\langle \ v_{c,i} \ , P\cdot \right\rangle) \ \ \| \leqslant \| \ Q - P^{\intercal}Q_c P \ \ \|.$$

If we could bound this term by a small quantity, we would be done. In 'Andreas Loukas, Graph reduction with spectral and cut guarantees' such an alignment between the eigenspaces of L and the lifted eigenspaces of L_c is attacked from the direction of canonical angles. This uses machinery introduced in Davis & Kahan (1970).

The canonical angle operator introduced there (and utilized in Loukas (2019) is defined as

$$\Theta = \begin{pmatrix} \Theta_0 & 0 \\ 0 & \Theta_1 \end{pmatrix}$$

with Θ_0, Θ_1 defined in eq. (1.16) of There it is then established (c.f. ibid. page 10) that $\| \ Q\breve{\ } P^{\intercal}Q_c P \ \ \| = \|\sin(\Theta)\|$. Hence, had we bounds on the entirety of Θ, we would be done. In 'Andreas Loukas, Graph reduction with spectral and cut guarantees', a bound on Θ_0 is provided (c.f. ibid. Theorem 14). However, without an additional bound on Θ_1 (c.f. Davis & Kahan (1970). eq. (1.16)) we unfortunately can not achieve our desired bound above.

E STABILITY WHEN $\|L - \tilde{L}\| \ll 1$

In this section we dicuss in addition to results in the main paper also stability in the setting where $\|L - \tilde{L}\| \ll 1$ as briefly considered at the beginning of Section 3. This is an important and well studied setting (Gama et al., 2019; 2020; Levie et al., 2019b; Kenlay et al., 2021b). It is different from the one considered in Section 4, as filter outputs are bounded with respect to a different notion of distance (i.e. the spectral difference $\|L - \tilde{L}\|$) than the notion of diffusion similarity.

We first reduce the transferability of entire networks to the transferability of basisi functions $\{psi_i\}_i$ making up the basis set Ψ of a given spectral convolutional network (c.f. Section 2).

Theorem E.1. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep graph convolutional architecture. Assume in each layer $1 \leqslant \ell \leqslant K$ that $\sum_i \|W_i^{\ell}\| \leqslant W$ and $\|B^{\ell}\| \leqslant B$. Choose $C \geqslant \|\Psi_i(L)\|$ $(\forall i \in I)$ and w.l.o.g. assume $CW > 1$. With this, we have with $\delta = \max_{i \in I}\{\|\Psi_i(L) - \Psi_i(\widetilde{L})\|\}$ that

$$\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, X)\| \leqslant \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW - 1}B\right)\right] \cdot \delta.$$

Proof. For simplicity in notation, let us denote the hidden representations in the network corresponding to $\tilde{L}$ by X^ℓ. With this, we note:

$$\begin{aligned}
\|X^K - \tilde{X}^K\| &\leqslant \sum_{i\in I} \|\psi_i(L) - \psi_i(\tilde{L})\| \cdot \|X^{K-1}\| \cdot \|W_i^K\| + \sum_{i\in I} \|\psi_i(\tilde{L})\| \cdot \|\tilde{X}^{K-1} - X^{K-1}\| \cdot \|W_i^K\| \\
&\leqslant \delta W \|X^{K-1}\| + CW\|\tilde{X}^{K-1} - X^{K-1}\| \\
&\leqslant \delta W \|X^{K-1}\| + CW\delta\|X^{K-2}\| + (CW)^2\|\tilde{X}^{K-1} - X^{K-1}\| \\
&\leqslant \frac{\delta}{C} \cdot \left(\sum_{\ell=1}^{K} (CW)^\ell \|X^{K-\ell}\|\right) \\
&= \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \|X^j\|\right)
\end{aligned}$$

Hence we need to bound the quantity $\|X^j\|$ in terms of C, W, B and X.

We have

$$\begin{aligned}
\|X^j\| &\leqslant \sum_i \|\psi_i(L)\| \cdot \|X^{j-1}\| \cdot \|W_i^j| + \|B^J\| \\
&\leqslant CW\|X^{j-1}\| + B \\
&\leqslant (CW)^2\|X^{j-2}\| + CWB + B \\
&\leqslant B\left(\sum_{k=0}^{j-1} (CW)^k\right) + (CW)^j\|X\| \\
&= \begin{cases} B\frac{(CW)^j - 1}{CW - 1} + (CW)^j\|X\| & ; CW \neq 1 \\ jB + \|X\| & ; CW = 1 \end{cases}.
\end{aligned}$$

For the case $CW = 1$, we thus find

$$\begin{aligned}
\|X^K - \tilde{X}^K\| &\leqslant \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (jB + \|X\|)\right) \\
&= \frac{\delta}{C} \cdot \left(K\|X\| + B\frac{K(K-1)}{2}\right).
\end{aligned}$$

For the case $CW \neq 1$, we find

$$\|X^K - \tilde{X}^K\| \leqslant \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \left[B\frac{(CW)^j - 1}{CW - 1} + (CW)^j\|X\|\right]\right)$$

For $CW > 1$, we may further estimate this as

$$\begin{aligned}
\|X^K - \tilde{X}^K\| &\leqslant \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \left[B\frac{(CW)^j - 1}{CW - 1} + (CW)^j\|X\|\right]\right) \\
&\leqslant \delta \cdot \frac{K(CW)^K}{C}\left[\frac{B}{CW - 1} + \|X\|\right].
\end{aligned}$$

This proves the claim. □

Theorem E.1 reduces the question of stability of entire networks to the question of *single filter stability* of the basis elements ψ_i in $\Psi = \{\psi_i\}_{i\in I}$. In practice, the difference "$\|\psi_i(L) - \psi_i(\widetilde{L})\|$" may of course be evaluated numerically if the basis Ψ is already given.

When *designing* new architectures, it is however important to know in advance how the choice of basis functions affects the stability properties of the network. To this end, bounds of the form

$\|\psi_i(L) - \psi_i(\widetilde{L})\| \leqslant C_{\psi_i} \cdot \|L - \widetilde{L}\|$ are desirable. Many existing works focus on deriving bounds of exactly this form (Gama et al., 2019; 2020; Levie et al., 2019b; Kenlay et al., 2021a;b).

Beyond this existing literature, we here provide an additional bound of the above form under the assumptions that $L, \tilde{L}$ are diagonalizable. This is always true for undirected graphs. Additionally, any Laplacian of a directed graph can be approximated by diagonalizable matices to arbitrary precision.

The bound below is based on existing work of Wihler (2009) who considered the case of *unitarily* diagonalizable matrices. To extend this to arbitrarily diagonalizable operators $L = V^{-1}\Lambda V$ we measure the severity of the failure to be *unitarily* diagonalizable via the **condition number** $\kappa(V_L) = \|V_L\| \cdot \|V_L^{-1}\|$ of the change-of-basis matrix V_L (with $\kappa(V_L) = 1$ whenever the change-of-basis matrix V_L is unitary).

Importantly in contrast to existing works, it should be noted that below we estimate the difference $\|\psi_i(L) - \psi_i(\widetilde{L})\|$ (which is measured in spectral norm $\|\cdot\|$) by the difference $\|L - \tilde{L}\|_F$ which is measured in *Frobenius* norm. Using the Frobenius norm as opposed to the spectral norm allows us to derive a uniform bound, where the the stability constant L_ψ does not depend on the eigenvalue structure of the respective Lapalcians $L, \tilde{L}$:

Theorem E.2. If $L, \tilde{L}$ are diagonalizable, we have with the Frobenius norm denoted by $\|\cdot\|_F$ that $\|\psi(\widetilde{L}) - \psi(L)\| \leqslant \kappa(V_L) \cdot \kappa(V_{\tilde{L}}) \cdot L_\psi \cdot \|\widetilde{L} - L\|_F$. Here L_{ψ_i} is the Lipschitz constant of ψ_i.

Proof. The claim directly follows from Lemma E.3 after noting that

$$||X||_{op} = \lambda_{\max}(X) \leqslant \sqrt{\sum_{i=1}^{n} \lambda_i^2(X)} = ||X||_F$$

□

Lemma E.3. Let $g : \mathbb{C} \to \mathbb{C}$ be Lipschitz continuous with Lipschitz constant D_g. Let X and Y satisfy

$$V^{-1}XV = \text{diag}(\lambda_1, ...\lambda_N) =: D(X)$$
$$W^{-1}YW = \text{diag}(\mu_1, ...\mu_N) =: D(Y).$$

This implies

$$\|g(X) - g(Y)\|_F \leqslant \|V^{-1}\|\|V\|\|W^{-1}\|\|W\| \cdot D_g \cdot \|X - Y\|_F.$$

Proof. This proof builds on the proof idea in Wihler (2009). We find:

$$\begin{aligned}
\|g(X) - g(Y)||_F^2 &= ||g(VD(X)V^{-1}) - g(WD(Y)W^{-1})\|_F^2 \\
&= \|Vg(D(X))V^{-1} - Wg(D(Y))W^{-1}\|_F^2 \\
&\leqslant \|V\|\|W^{-1}\| \cdot \|g(D(X))V^{-1}W - V^{-1}Wg(D(Y))\|_F^2 \\
&= \|V\|\|W^{-1}\| \cdot \sum_{i,j} \left|(g(D(X))V^{-1}W - V^{-1}Wg(D(Y)))_{ij}\right|^2 \\
&= \|V\|\|W^{-1}\| \cdot \sum_{i,j} \left|\sum_k [g(D(X))]_{ik}[V^{-1}W]_{kj} - [V^{-1}W]_{ik}[g(D(Y))]_{kj}\right|^2 \\
&= \|V\|\|W^{-1}\| \cdot \sum_{i,j} \left|[V^{-1}W]_{ij}\right|^2 |g(\lambda_j) - g(\mu_i)|^2 \\
&\leqslant \|V\|\|W^{-1}\| \cdot \sum_{i,j} \left|[V^{-1}W]_{ij}\right|^2 D_g^2 |\lambda_j - \mu_i|^2 \\
&= \|V\|\|W^{-1}\| \cdot D_g^2 \|D(X)V^{-1}W - V^{-1}WD(Y)\|_F^2 \\
&\leqslant \|V\|\|V^{-1}\|\|W^{-1}\|\|W\| \cdot D_g^2 \|X - Y\|_F^2.
\end{aligned}$$

□

F Comparison of Diffusion Flows for edge-rewiring in K_N

We are interested in establishing that in the setting of Section 3, we have

$$\|e^{-Lt} - e^{-\tilde{L}t}\| \lesssim e^{-(N-2)t}.$$

To this end, we first note that both Laplacians $L, \tilde{L}$ correspond to graphs that are connected. Hence the kernel of both Laplacians is spanned by the vector of $\mathbb{1}$ of all ones. Denote by P the orthogonal projection onto $\mathbb{1}$ and set $Q = Id - P$. We then have

$$\|e^{-Lt} - e^{-\tilde{L}t}\| = \|Qe^{-Lt}Q - Qe^{-\tilde{L}t}Q\|.$$

Next we note for the Laplacian L on K_N that

$$L = N \cdot Q,$$

and hence

$$\|e^{-Lt} - e^{-\tilde{L}t}\| = \|Qe^{-Nt} - Qe^{-\tilde{L}t}Q\|.$$

From perturbation theory, we note that for the eigenvalues of symmetric matrices $A, (A + B)$ ordered in decreasing order, we have (c.f. e.g. Kato (1976))

$$|\lambda_i(A + B) - \lambda_i(A)| \leqslant \|B\|.$$

Since $\tilde{L}$ arises from L by deleting a single edge and the Laplacian defined on an unweighted connected two-node graph has operator norm equal to two, we find

$$|\lambda - N| \leqslant 2$$

for any $\lambda \in \sigma(\tilde{L})$. Thus with spectral projection P_λ of $\tilde{L}$, we find

$$\|e^{-Lt} - e^{-\tilde{L}t}\| \leqslant e^{-Nt} \left\| \sum_{0 \neq \lambda \in \sigma(\tilde{L})} Q(1 - e^{(N-\lambda)t} P_\lambda Q \right\| \lesssim e^{-(N-2)t}.$$

G Example of unidirectionally similar graphs

Here we further discuss the example of unidirectionally similar graphs introduced in Fig. 5 of Section 3.

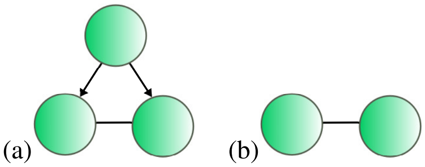

Figure 15: Example of unidirectionally similar graphs

Let us denote the graph of Fig. 15 (a) by $\tilde{G}$ and the graph of Fig. 15 (b) by G. On both these graphs let us consider the out-degree Laplacian (c.f 2.1)

$$L^{\text{out}} := D^{\text{out}} - W$$

as characteristic operator on both G and $\tilde{G}$.

The diffusion process e^{-tL} arises as the solution operator of the differential equation

$$\frac{dx(t)}{dt} = -Lx(t).$$

Using this, we see that no information flows from the 'top' node of $\tilde{G}$ to either of the two bottom nodes in Fig. 15 (a). Chosing as J the obvious inclusion operator mapping from $\tilde{G}$ to G and assigning the value '0' to the top node in $\tilde{G}$, we easily find $\|e^{-tL}J - e^{-t\tilde{L}}J\| = 0$. The diffusion on $\tilde{G}$ (i.e. the graph in Fig. 15 (a)) however is dependent on the top node in $\tilde{G}$ as well if this node carries a non-zero initial value. Hence we can not transfer it to G.

H Laplace Transform Filters

In this section we provide an overview of the concept of Laplace transforms. We begin with a recapitulation of complex measures.

H.1 Complex measures on $\mathbb{R}_{\geqslant 0}$ and their Theory of Integration

As reference for this section Tao (2013) might serve.

In mathematics, a measure is a formal generalization of concepts such as length, area and volume. We are interested in assigning a generalized notion of length (or mass) to subsets of the real half-line

$$\mathbb{R}_{\geqslant 0} = [0, \infty).$$

The set will turn out to be a so called σ-Algebra; i.e. a set Σ of sets for which

- $\varnothing, \mathbb{R}_{\geqslant 0} \in \Sigma$
- $A, B \in \sigma \Rightarrow A \cap B \in \Sigma$
- $A, B \in \Sigma \Rightarrow A \backslash B \in \Sigma$
- $A, B \in \Sigma \Rightarrow A \cup B \in \Sigma$.

We now take $\Sigma_{\mathbb{R}_{\geqslant 0}}$ to be the smallest such set of sets Σ that contains all open intervals.

A complex measure then is a set-function that assigns to each set in $\Sigma_{\mathbb{R}_{\geqslant 0}}$ a complex number in a certain way:

Definition H.1. A complex measure μ on $\mathbb{R}_{\geqslant 0}$ is a complex valued function $\mu : \Sigma_{\mathbb{R}_{\geqslant 0}} \rightarrow \mathbb{C}$ satisfying

$$\mu\left(\bigcup_n A_n\right) = \sum_n \mu\left(A_n\right)$$

for any countable (potentially infinite) collection of sets in $\Sigma_{\mathbb{R}_{\geqslant 0}}$ which are pairwise disjoint.

Let us provide some examples:

Example H.2. The prototypical example of a measure is the standard Lebesgue measure that assigns to any interval (a, b) the length $\mu_{\text{Leb}}((a, b)) = |a - b|$ $(a, b \in \mathbb{R}_{\geqslant 0})$.

Example H.3. Alternatively, we might consider the Dirac measure $\mu_{\delta_{t_0}}$, which assigns the value $\mu_{\delta_{t_0}}((a, b)) = 1$ to any interval (a, b) containing t_0 (i.e. $t_0 \in (a, b)$). Otherwise it assigns the value $\mu_{\delta_{t_0}}((a, b)) = 0$ if $t_0 \notin (a, b)$.

Example H.4. Every integrable function $\hat{\psi} : \mathbb{R}_{\geqslant 0} \rightarrow \mathbb{C}$ defines a complex measure via $\mu_{\hat{\psi}}((a, b)) = \int_a^b \hat{\psi}(t) dt$.

Any given measure on $\mathbb{R}_{\geqslant 0}$ defines a unique way of integrating (known as Lebesgue integration) a function f defined on $\mathbb{R}_{\geqslant 0}$. This proceeds by approximating any function f via a weighted sequence of indicator functions (with $A \in \Sigma_{\mathbb{R}_{\geqslant 0}}$ a set)

$$\chi_A(t) = \begin{cases} 1 & ; t \in A \\ 0 & ; t \notin A \end{cases}.$$

as

$$f(t) \approx f_n(t) := \sum_k a_k^n \chi_{A_k}(t).$$

with $a_k \in \mathbb{C}$. For these functions, one then sets

$$\int_{\mathbb{R}_{\geqslant 0}} f_n d\mu \equiv \sum_k a_k^n \cdot \mu(A_k).$$

Since we have $\lim_{n \rightarrow \infty} f_n = f$, one then simply sets

$$\int_{\mathbb{R}_{\geqslant 0}} f d\mu \equiv \lim_{n \rightarrow \infty} \int_{\mathbb{R}_{\geqslant 0}} f_n d\mu.$$

Example H.5. For the prototypical example of the standard Lebesgue measure, this process simply yields

$$\int_{\mathbb{R}_{\geqslant 0}} f(t) d\mu_{\text{Leb}}(t) = \int_0^\infty f(t) dt.$$

Example H.6. For the Dirac measure $\mu_{\delta_{t_0}}$, the above process yields

$$\int_{\mathbb{R}_{\geqslant 0}} f(t) d\mu_{\delta_{t_0}}(t) = f(t_0)$$

Example H.7. For measures arising from integrable functions $\hat{\psi} : \mathbb{R}_{\geqslant 0} \to \mathbb{C}$ as $\mu_{\hat{\psi}}((a,b)) = \int_a^b \hat{\psi}(t)dt$, we find

$$\int_{\mathbb{R}_{\geqslant 0}} f(t) d\mu_{\hat{\psi}} = \int_0^\infty \hat{\psi}(t) f(t) dt.$$

H.2 Laplace Transforms

We say complex valued measure μ is finite if we have

$$\int_{\mathbb{R}_{\geqslant 0}} d|\mu|(t) < \infty.$$

Here the measure $|\mu|$ arises from the original measure μ via

$$|\mu|((a,b)) \equiv |\mu((a,b))|.$$

For any such finite measure μ we may define its Laplace transform as

$$\psi_\mu(z) := \int_{\mathbb{R}_{\geqslant 0}} e^{-tz} d\mu(t).$$

This function f_μ is well defined for z in the right hemisphere

$$\mathbb{C}_R := \{z \in \mathbb{C} : \text{Re}(z) \geqslant 0\}.$$

of the complex plane $\mathbb{C}$, since there we have

$$\begin{aligned} |\psi_\mu(z)| &= \left| \int_{\mathbb{R}_{\geqslant 0}} e^{-tz} d\mu(t) \right| \\ &\leqslant \int_{\mathbb{R}_{\geqslant 0}} |e^{-tz}| d|\mu|(t) \\ &\leqslant \int_{\mathbb{R}_{\geqslant 0}} d|\mu|(t) < \infty. \end{aligned}$$

Example H.8. For the Dirac measure $\mu_{\delta_{t_0}}$, we have

$$\psi_{\mu_{\delta_{t_0}}}(z) = e^{-t_0 z}.$$

Example H.9. For any integrable function $\hat{\psi}$, we have

$$\psi(z) \equiv \int_{\mathbb{R}_{\geqslant 0}} e^{-tz} d\mu_{\hat{\psi}} = \int_0^\infty \hat{\psi}(t) e^{-tz} dt.$$

More specifically, if the integrable function is given as $\hat{\psi}_k := (-t)^{k-1} e^{-\lambda t}$ (with $\text{Re}(\lambda) > 0$), then $\psi_k(z) = (z + \lambda)^{-k}$:

Example H.10. If $\hat{\psi}_k := (-t)^{k-1} e^{-\lambda t}$ yields $\psi_k(z) = (z + \lambda)^{-k}$, then

$$\psi_k(z) = (z + \lambda)^{-k}.$$

For $k = 1$, this can be seen from

$$\int_0^\infty e^{-tz} e^{-\lambda t} dt = -\frac{1}{z + \lambda} e^{-(z+\lambda)} \Big|_0^\infty.$$

For $k > 1$, the claim follows from differentiating the above expression with respect to z Note that the functions $\psi_k(z) = (z + \lambda)^{-k}$ are also defined if $\text{Re}(z) \leqslant 0$, as long as $z \neq -\lambda$.

Using the function ψ_k of the examples above, a wide class of functions may be parametrized

Theorem H.11. *Let $f : \mathbb{R}_{\geqslant 0} \to 0$ be any function with $\lim_{x\to\infty} f(x) = 0$. Then for any $\epsilon > 0$, there is a function*

$$h(x) = \sum_k \theta_k \psi_k(x)$$

for which

$$\sup_{x\in[0,\infty)} |f(x) - h(x)| < \epsilon.$$

Here the basis functions $\{\psi_k\}$ may either be chosen as $\psi_k(z) = (z+\lambda)^{-k}$ or $\psi_k(x) = e^{-(kt_0)x}$ for any $t_0 > 0$.

Proof. This is a direct consequence of the Weierstrass approximation theorem. □

H.3 Proof of Theorem 4.4

In this section, we prove Theorem 4.4, which we restate here for convenience:

Theorem H.12. We have $\|J\psi(L) - \psi(\tilde{L})J\| \leqslant \|\hat{\psi}\|_1 \cdot \sup_{t\geqslant 0} \|Je^{-Lt} - e^{-\tilde{L}t}J\|$ in the *unidirectional* setting. In the *bidirectional* setting $\|\psi(L) - \tilde{J}\psi(\tilde{L})J\| \leqslant \int_0^\infty |\hat{\psi}(t)|\eta(t)dt$ holds true.

Proof. We start by proving the first claim. To this end, we note

$$\begin{aligned}\|J\psi(L) - \psi(\tilde{L})J\| &= \left\| \int_{\mathbb{R}_{\geqslant 0}} \left[Je^{-tL} - e^{-t\tilde{L}}J \right] d\mu_{\hat{\psi}} \right\| \\ &\leqslant \int_{\mathbb{R}_{\geqslant 0}} \left\| \left[Je^{-tL} - e^{-t\tilde{L}}J \right] \right\| d|\mu|_{\hat{\psi}} \\ &\leqslant \sup_{t\geqslant 0} \|Je^{-Lt} - e^{-\tilde{L}t}J\| \cdot \int_{\mathbb{R}_{\geqslant 0}} d|\mu|_{\hat{\psi}}\end{aligned}$$

Observing that in the notation of Section 4.2 we precisely have

$$\|\hat{\psi}\|_1 \equiv \int_{\mathbb{R}_{\geqslant 0}} d|\mu|_{\hat{\psi}}$$

the claim follows.

Proceeding as above, we note

$$\|\psi(L) - \tilde{J}\psi(\tilde{L})J\| \leqslant \int_0^\infty \left\| \left[e^{-tL} - \tilde{J}e^{-t\tilde{L}}J \right] \right\| d|\mu|_{\hat{\psi}},$$

from which the second claim follow.

□

H.4 Proof of Corollary 4.5

Here we prove Corollary 4.5; restated here for convenience:

Corollary H.13. *Consider a sequence of graphs G_n for which $\|e^{-L_n t} - \tilde{J}_n e^{-\tilde{L}t} J_n\| \to 0$. Then for a Laplace transform filter ψ, we have $\|\psi(L_n) - \tilde{J}_n\psi(\tilde{L})J_n\|| \to 0$ if and only if $\lim_{r\to\infty} \psi(r) = 0$.*

Proof. Let us first prove that the condition is sufficient. To this end assume that $\lim_{r\to\infty} \psi(r) = 0$. This implies that $\mu_{\hat{\psi}}(\{0\}) = 0$. Hence we have

$$\begin{aligned}\|\psi(L_n) - \tilde{J}_n\psi(\tilde{L})J_n\| &= \left\| \int_0^\infty \left[e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J \right] d\mu_{\hat{\psi}}(t) \right\| \\ &\leqslant \int_0^\infty \left\| e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J \right\| d|\mu|_{\hat{\psi}}(t)\end{aligned}$$

The integrand $\left\| e^{-Lt} - \tilde{J}e^{-\tilde{L}t}J \right\|$ converges to zero everywhere except on a set of measure zero (i.e. the set $\{t|t = 0\} = \{0\}$). The dominated convergence theorem then yields the claim. □

H.5 Additional Technical Convergence Result for Laplace Transform Filters

Here we prove an additional technical convergence result, which will be needed in section I.1.

For a generic operator, we measure the failure to be (unitarily) diagonalizable via its so-called **departure from normality** $\nu^2(L) = (\|L\|_F^2 - \sum_{\lambda_k \in \sigma(L)} |\lambda_k|^2)$ which is zero if and only if L is unitarily diagonalizable Bandtlow (2004).
We then have:

Theorem H.14. Let ψ be a Laplace transform filter. There exists a constant $C = C_{\psi,\nu(L),\nu(\tilde{L})} < \infty$ so that we have $\|J\psi(L) - \psi(\tilde{L})J\| \leqslant C \cdot \|J(L + \lambda Id)^{-1} - (\tilde{L} + \lambda \tilde{Id})^{-1}J\|$.

Proof. We make use of the holomorphic functional calculus (Kato, 1976; Post, 2012) to represent $\psi(L)$ as

$$\psi(L) := -\frac{1}{2\pi i}\oint\limits_{\Gamma} \psi(z) \cdot (L - z \cdot Id)^{-1} dz$$

to arrive at

$$\|J\psi(L) - \psi(\tilde{L})J\| \leqslant \frac{1}{2\pi}\oint\limits_{\Gamma} |\psi(z)| \cdot \|J(L - zId)^{-1} - (\tilde{L} - zId)^{-1}J\| d|z|.$$

Combining results of Post (2012) and Bandtlow (2004) yields

$$\begin{aligned}
&\|J(L - zId)^{-1} - (\tilde{L} - zId)^{-1}J\| \\
\leqslant &\left(1 + |\lambda + z|\frac{\sqrt{e}}{d(z,\sigma(\tilde{L}))}\exp\left(\frac{1}{2}\frac{\nu(\tilde{L})}{d(z,\sigma(\tilde{L}))}\right)\right) \cdot \left(1 + |\lambda + z|\frac{\sqrt{e}}{d(z,\sigma(\tilde{L}))}\exp\left(\frac{1}{2}\frac{\nu(\tilde{L})}{d(z,\sigma(\tilde{L}))}\right)\right) \\
\times &\|J(L + \lambda Id)^{-1} - (\tilde{L} + \lambda Id)^{-1}J\|.
\end{aligned}$$

Hence we may set

$$C = \frac{1}{2\pi}\oint\limits_{\Gamma} |\psi(z)| \cdot p_{\nu(L),\nu(\tilde{L})}(z) d|z|$$

with

$$\begin{aligned}
&p_{\nu(L),\nu(\tilde{L})}(z) \\
\equiv &\left(1 + |\lambda + z|\frac{\sqrt{e}}{d(z,\sigma(\tilde{L}))}\exp\left(\frac{1}{2}\frac{\nu(\tilde{L})}{d(z,\sigma(\tilde{L}))}\right)\right) \cdot \left(1 + |\lambda + z|\frac{\sqrt{e}}{d(z,\sigma(\tilde{L}))}\exp\left(\frac{1}{2}\frac{\nu(\tilde{L})}{d(z,\sigma(\tilde{L}))}\right)\right)
\end{aligned}$$

□

Such a result also holds in the bidirectional setting:

Theorem H.15. Consider a graph sequence G_n with $\|(L_n + \lambda Id)^{-1} - \tilde{J}_n(\tilde{L} + \lambda Id)^{-1}J_n\| \to 0$. If the graphs are directed, assume eigenvalues of all L_ns lie within a cone of opening angle $\alpha < \pi$ symmetric about the real axis. Then we have $\|\psi(L_n) - \tilde{J}_n\psi(\tilde{L})J_n\| \to 0$ if and only if $\lim_{r\to\infty}\psi(r) = 0$.

Proof. As in the proof above, we arrive at

$$\|\psi(L) - \tilde{J}\psi(\tilde{L})J\| \leqslant \frac{1}{2\pi}\oint\limits_{\Gamma} |\psi(z)| \cdot \|(L - zId)^{-1} - \tilde{J}(\tilde{L} - zId)^{-1}J\| d|z|.$$

Since $\|(L_n + \lambda Id)^{-1} - \tilde{J}_n(\tilde{L} + \lambda Id)^{-1}J_n\| \to 0$ implies $\|(L_n - zId)^{-1} - \tilde{J}_n(\tilde{L} - zId)^{-1}J_n\| \to 0$ uniformly (in z) on compact sets (c.f. e.g. Arendt (2001)), we can apply dominated convergence as in the proof of Corollary 4.5 in Appendix H.4; if we find an majorizing function that is integrable on Γ. But this is ensured by the decay of ψ and the possibility to choose Γ to lie within in a cone of opening angle $\alpha \lneq \pi$ about the real axis of opening angle less than π. □

H.6 Discussion of extension beyond spectral assumptions

Above, we have assumed that all appearing eigenvalues $\lambda \in \mathbb{C}$ in the spectrum $\sigma(L)$ have real part $\mathrm{Re}(\lambda) \geqslant 0$. This guarantees that

$$\limsup_{t\rightarrow\infty} \|e^{-Lt}\| < \infty.$$

From this we find that

$$\|\psi(L)\| = \left\|\int_{\mathbb{R}_{\geqslant 0}} e^{-tL} d\mu(t)\right\| \leqslant \left(\limsup_{t\rightarrow\infty} \|e^{-Lt}\|\right) \cdot \int_{\mathbb{R}_{\geqslant 0}} d|\mu|(t) < \infty,$$

so that the filter $\psi(L)$ is indeed well-defined. If we want to allow $\mathrm{Re}(\lambda) < 0$ as well, we have two options:

The set $\{\mathbf{Re}(\lambda)\}$ is bounded from below: In this setting we have a guarantee that there is $c_- > 0$ so that for all appearing eigenvalues in the spectra of L and $\tilde{L}$ we have

$$-c_- \leqslant \mathrm{Re}(\lambda).$$

This implies that

$$\limsup_{t\rightarrow\infty} \|e^{-Lt}e^{-c_- t}\| < \infty.$$

Using

$$\left\|\int_{\mathbb{R}_{\geqslant 0}} e^{-tL} d\mu(t)\right\| = \left\|\int_{\mathbb{R}_{\geqslant 0}} e^{-tL}e^{-c_- t}e^{c_- t} d\mu(t)\right\|$$
$$\leqslant \left(\limsup_{t\rightarrow\infty} \|e^{-Lt}e^{-c_- t}\|\right) \cdot \int_{\mathbb{R}_{\geqslant 0}} e^{c_- t} d|\mu|(t),$$

the developed theory above is still applicable in this setting, as long as we assume that the measure μ defining the Laplace transform filter ψ satisfies

$$\int_{\mathbb{R}_{\geqslant 0}} e^{c_- t} d|\mu|(t) < \infty.$$

Note that this is stronger than the demand

$$\int_{\mathbb{R}_{\geqslant 0}} d|\mu|(t) < \infty.$$

made in Definition 4.1.

The set $\{\mathbf{Re}(\lambda)\}$ is not bounded from below: In this setting, we pick a $\mu \in \mathbb{C}$ with $\mathrm{Re}(\mu) < 0$ and $\mu \notin \sigma(L) \cup \sigma(\tilde{L})$. We then restrict the class of filters to those determined by Example 4.3: There we chose $\hat{\psi}_k := (-t)^{k-1}e^{-\mu t}$, which yielded filters of the form $\{h_\theta(\cdot) := \sum_i \theta_i \cdot \psi_i(\cdot)\}$, with $\psi_k(L) = \left[(L + \mu Id)^{-1}\right]^k$. Such filters hence remain defined as long as $\mu \notin \sigma(L)$.

H.7 Proof of Theorems 4.6 & 4.7

Theorem H.16. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep LTF-based network. Assume $\sum_{i\in I} \|W_i^\ell\| \leqslant W$ and $\|B^\ell\| \leqslant B$. Choose $C \geqslant \|\Psi_i(\tilde{L})\|$ $(i \in I)$ and w.l.o.g. assume $CW > 1$. Assume $\rho(J\tilde{X}) = J\rho(\widetilde{X})$. If biases are enabled, assume $J\mathbb{1}_G = \mathbb{1}_{\tilde{G}}$. Then we have with $\delta = \max_{i\in I}\{\|J\psi_i(L) - \psi_i(\widetilde{L})J\|\}$:

$$\|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\tilde{L}, JX)\| \leqslant \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1}B\right)\right] \cdot \delta.$$

Proof. Let us define

$$\tilde{X} := JX.$$

Let us further use the notation $\tilde{\psi}_i := \psi_i(\tilde{L})$ and $\psi_i := \psi_i(L)$.

Denote by X^ℓ and $\widetilde{X}^\ell$ the (hidden) feature matrices generated in layer ℓ for networks based on ψ_i and $\tilde{\psi}_i$ respectively: I.e. we have

$$X^\ell = \rho\left(\sum_{i \in I} \psi_i X^{\ell-1} W_i^\ell + B^\ell\right)$$

and

$$\widetilde{X}^\ell = \rho\left(\sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{\ell-1} W_i^\ell + \tilde{B}^\ell\right).$$

We then have

$$\begin{aligned}
&\|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \\
=&\|JX^K - \widetilde{X}^K\| \\
=&\left\|J\rho\left(\sum_{i \in I} \psi_i X^{L-1} W_i^K + B^K\right) - \rho\left(\sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{K-1} W_i^K + \tilde{B}^L\right)\right\| \\
=&\left\|\rho\left(J\sum_{i \in I} \psi_i X^{L-1} W_i^K + \tilde{B}^K\right) - \rho\left(\sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{K-1} W_i^K + B^L\right)\right\|
\end{aligned}$$

Here we used the assumption that ρ and J commute. We also made use of the assumption $J\mathbb{1}_G = \mathbb{1}_{\tilde{G}}$ when dealing with biases .
Using the fact that $\rho(\cdot)$ is 1-Lipschitz-continuous (c.f. Section 2.3), we can establish

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \\
\leqslant&\left\|\left(J\sum_{i \in I} \psi_i X^{L-1} W_i^K + \tilde{B}^K\right) - \left(\sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{K-1} W_i^K + \tilde{B}^K\right)\right\|.
\end{aligned}$$

We then have

$$\begin{aligned}
&\|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \\
\leqslant&\left\|\sum_{i \in I} J\psi_i X^{K-1} W_i^K - \sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{K-1} W_i^K\right\|.
\end{aligned}$$

From this, we find (inserting a zero), that

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \\
\leqslant&\left\|\sum_{i \in I} J\psi_i X^{K-1} W_i^K - \sum_{i \in I} \tilde{\psi}_i \widetilde{X}^{K-1} W_i^K\right\| \\
\leqslant&\left\|\sum_{i \in I} (J\psi_i - \tilde{\psi}_i J) X^{K-1} W_i^K\right\| + \sum_{i \in I} \|\tilde{\psi}_i\| \cdot \|\widetilde{X}^{K-1} - JX^{K-1}\| \cdot \|W_i^K\| \\
\leqslant&\left\|\sum_{i \in I} (J\psi_i - \tilde{\psi}_i J) X^{K-1} W_i^K\right\| + CW \cdot \|\widetilde{X}^{K-1} - JX^{K-1}\| \\
\leqslant&\sum_{i \in I} \left\|(J\psi_i - J\tilde{\psi}_i J)\right\| \cdot \|X^{K-1}\| \cdot \|W_i^K\| + CW \cdot \|\widetilde{X}^{K-1} - JX^{K-1}\| \\
\leqslant&\sum_{i \in I} \delta \cdot \|X^{K-1}\| W + CW \cdot \|\tilde{J}\widetilde{X}^{K-1} - X^{K-1}\|
\end{aligned}$$

Arguing as in the proof of Appendix E then yields the claim.

□

For the bidirectional setting we find the following:

Theorem H.17. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep LTF-based network. Assume that $\sum_{i\in I}\|W_i^\ell\| \leqslant W$ and $\|B^k\| \leqslant B$. Choose $C \geqslant \|\Psi_i(L)\|, \|\Psi_i(\tilde{L})\|$ $(i \in I)$ and w.l.o.g. assume $CW > 1$. Assume $\rho(\tilde{J}X) = \tilde{J}\rho(X)$ and if biases are enabled, assume $\tilde{J}\mathbb{1}_{\tilde{G}} = \mathbb{1}_G$. Set $\max_{i\in I}\{\|\psi_i(L) - \tilde{J}\psi_i(\tilde{L})J\|\} = \delta_1$ and define $\delta_2 = \max_{i\in I}\{\|\psi_i(\tilde{L})[J\tilde{J} - Id_{\tilde{G}}]\|\}$. With this, we have that

$$\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \leqslant \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1}B\right)\right] \cdot (\delta_1 + \delta_2).$$

Proof. Let us define

$$\tilde{X} := JX.$$

Let us further use the notation $\tilde{\psi_i} := \psi_i(\tilde{L})$ and $\psi_i := \psi_i(L)$.

Denote by X^ℓ and $\widetilde{X}^\ell$ the (hidden) feature matrices generated in layer ℓ for networks based on ψ_i and $\tilde{\psi_i}$ respectively: I.e. we have

$$X^\ell = \rho\left(\sum_{i\in I}\psi_i X^{\ell-1}W_i^\ell + B^\ell\right)$$

and

$$\widetilde{X}^\ell = \rho\left(\sum_{i\in I}\tilde{\psi_i}\widetilde{X}^{\ell-1}W_i^\ell + \tilde{B}^\ell\right).$$

We then have

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \\
=&\|X^K - \tilde{J}\widetilde{X}^K\| \\
=&\left\|\rho\left(\sum_{i\in I}\psi_i X^{K-1}W_i^K + B^K\right) - \tilde{J}\rho\left(\sum_{i\in I}\tilde{\psi_i}\widetilde{X}^{K-1}W_i^K + \tilde{B}^L\right)\right\| \\
=&\left\|\rho\left(\sum_{i\in I}\psi_i X^{K-1}W_i^K + B^K\right) - \rho\left(\tilde{J}\sum_{i\in I}\tilde{\psi_i}\widetilde{X}^{K-1}W_i^K + B^L\right)\right\|
\end{aligned}$$

Here we used the assumption that ρ and $\tilde{J}$ commute. fact that since ReLU$(\cdot)$ maps positive entries to positive entries and acts pointwise, it commutes with $J^\uparrow$. We also made use of the assumption $\tilde{J}\mathbb{1}_{\tilde{G}} = \mathbb{1}_G$ when dealing with biases .
Using the fact that $\rho(\cdot)$ is 1-Lipschitz-continuous (c.f. Section 2.3), we can establish

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \\
\leqslant&\left\|\rho\left(\sum_{i\in I}\psi_i X^{K-1}W_i^K + B^K\right) - \rho\left(\tilde{J}\sum_{i\in I}\tilde{\psi_i}\widetilde{X}^{K-1}W_i^K + B^L\right)\right\| \\
\leqslant&\left\|\sum_{i\in I}\psi_i X^{K-1}W_i^K + B^K - \tilde{J}\sum_{i\in I}\tilde{\psi_i}\widetilde{X}^{K-1}W_i^K + B^K\right\|.
\end{aligned}$$

Using the assumption that $\|\tilde{\psi}[J\tilde{J} - Id_{\tilde{G}}]\| \leqslant \delta_2$, we have

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \\
\leqslant &\left\|\sum_{i\in I}\psi_i X^{K-1}W_i^K - \sum_{i\in I}(\tilde{J}\tilde{\psi}_i J)\tilde{J}\widetilde{X}^{K-1}W_i^K\right\| + \left\|\sum_{i\in I}\tilde{J}\tilde{\psi}_i[Id_{\tilde{G}} - J\tilde{J}]\widetilde{X}^{K-1}W_i^K\right\| \\
\leqslant &\left\|\sum_{i\in I}\psi_i X^{K-1}W_i^K - \sum_{i\in I}(\tilde{J}\tilde{\psi}_i J)\tilde{J}\widetilde{X}^{K-1}W_i^K\right\| + \delta_2\cdot\left\|\sum_{i\in I}\widetilde{X}^{K-1}W_i^K\right\| \\
\leqslant &\left\|\sum_{i\in I}\psi_i X^{K-1}W_i^K - \sum_{i\in I}(\tilde{J}\tilde{\psi}_i J)\tilde{J}\widetilde{X}^{K-1}W_i^K\right\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W
\end{aligned}$$

From this, we find (assuming $\|\tilde{J}\|, \|J\| \leqslant 1$), that

$$\begin{aligned}
&\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \\
\leqslant &\left\|\sum_{i\in I}\psi_i X^{K-1}W_i^K - \sum_{i\in I}(\tilde{J}\tilde{\psi}_i J)\tilde{J}\widetilde{X}^{K-1}W_i^K\right\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W \\
\leqslant &\left\|\sum_{i\in I}(\psi_i - \tilde{J}\tilde{\psi}_i J)X^{K-1}W_i^K\right\| + \sum_{i\in I}\|\tilde{J}\tilde{\psi}_i J\|\cdot\|\tilde{J}\widetilde{X}^{K-1} - X^{K-1}\|\cdot\|W_i^K\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W \\
\leqslant &\left\|\sum_{i\in I}(\psi_i - \tilde{J}\tilde{\psi}_i J)X^{K-1}W_i^K\right\| + CW\cdot\|\tilde{J}\widetilde{X}^{K-1} - X^{K-1}\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W \\
\leqslant &\sum_{i\in I}\left\|(\psi_i - \tilde{J}\tilde{\psi}_i J)\right\|\cdot\left\|X^{K-1}\right\|\cdot\left\|W_i^K\right\| + CW\cdot\|\tilde{J}\widetilde{X}^{K-1} - X^{K-1}\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W \\
\leqslant &\delta_1\cdot\left\|X^{K-1}\right\|W + CW\cdot\|\tilde{J}\widetilde{X}^{K-1} - X^{K-1}\| + \delta_2\cdot\left\|\widetilde{X}^{K-1}\right\|\cdot W
\end{aligned}$$

Arguing as in the proof of Appendix E then yields the claim.

Discussion of the condition $\delta_2 = \max_{i\in I}\{\|\psi_i(\tilde{L})[J\tilde{J} - Id_{\tilde{G}}]\|\} \ll 1$ Since $\lim_{r\to\infty}\psi_i(r) = 0$, $J\tilde{J}$ only needs to map eigenvectors of L corresponding to small eigenvalues approximately to themselves. On the remaining eigenvectors, $\psi_i(L)$ will already approximately act as zero. Since only one of the factors in the product $\psi_i(\tilde{L})\cdot[J\tilde{J} - Id_{\tilde{G}}]$ needs to be approximately zero, this relaxes conditions on how the remaining factor (i.e. $[J\tilde{J} - Id_{\tilde{G}}]$) needs to act on such eigenvectors.

□

H.8 Proof of Theorem 4.9

Here we prove Theorem 4.9; restated again for convenience:

Theorem H.18. Assuming $\Omega(JX) = \Omega(X)$, we have in the setting of Theorem 4.6 that $\|\Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \leqslant \|J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\|$.
Assuming $\Omega(\tilde{X}) = \Omega(\tilde{J}\tilde{X})$, we have in the (bidirectional) setting of Theorem 4.7 that $\|\Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \leqslant \|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\|$.

Proof. We note

$$\begin{aligned}
&\|\Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X) - \Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX)\| \\
=&\|\Omega(\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X)) - \Omega(\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX))\| \\
=&\|\Omega(J\Phi_{\mathscr{W},\mathscr{B},\Psi}(L,X)) - \Omega(\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L},JX))\|.
\end{aligned}$$

To prove the claim from here, we only have to note that the aggregation method Ω as defined in Section 4.3.2 is 1-Lipschitz (as a consequence of the reverse triangle inequality). The proof for the bidirectional setting proceeds analogously. □

A similar proof shows the following for the bidirectional setting:

Theorem H.19. Assuming $\Omega(X) = \Omega(\tilde{J}X)$, we have in the setting of Theorem H.17 that $\|\Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\| \leqslant \|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \tilde{J}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\widetilde{L}, JX)\|$.

I Further Discussion for Examples of Transferability Settings

I.1 Further Discussion of the setting of coarse-graining Graphs

In this appendix, we illustrate:

$$\|(\Delta + Id)^{-1} - J^{\uparrow}(\underline{\Delta} + Id)^{-1}J^{\downarrow}\| \lesssim 1/\lambda_1(\Delta_{\text{high}}).$$

Using Theorem H.15, then yields the prove of the desired equality (3)

$$\|e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\| \lesssim 1/w^{\text{min}}_{\text{high}} \;\text{ for any }\; t > 0.$$

after noting the linear relation in scaling behaviour $\lambda_1(L_{\text{cluster}}) \sim w^{\text{min}}_{\text{high}}$.

For convenience, we restate the definitions leading up to this setting again:

Definition I.1. Denote by $\underline{\mathcal{G}}$ the set of connected components in G_{high}. We give this set a graph structure as follows: Let R and P be elements of $\underline{\mathcal{G}}$ (i.e. connected components in G_{high}). We define the real number

$$\underline{W}_{RP} = \sum_{r \in R} \sum_{p \in P} W_{rp},$$

with r and p nodes in the original graph G. We define the set of edges $\underline{\mathcal{E}}$ on $\underline{G}$ as

$$\underline{\mathcal{E}} = \{(R, P) \in \underline{\mathcal{G}} \times \underline{\mathcal{G}} : \underline{W}_{RP} > 0\}$$

and assign $\underline{W}_{RP}$ as weight to such edges. Node weights of limit nodes are defined similarly as aggregated weights of all nodes r (in G) contained in the component R as

$$\underline{\mu}_R = \sum_{r \in R} \mu_r.$$

In order to translate signals between the original graph G and the limit description $\underline{G}$, we need translation operators mapping signals from one graph to the other:

Definition I.2. Denote by $\mathbb{1}_R$ the vector that has 1 as entries on nodes r belonging to the connected (in G_{hign}) component R and has entry zero for all nodes not in R. We define the down-projection operator $J^{\downarrow}$ component-wise via evaluating at node R in $\underline{\mathcal{G}}$ as

$$(J^{\downarrow}x)_R = \langle \mathbb{1}_R, x \rangle / \underline{\mu}_R.$$

The upsampling operator $J^{\uparrow}$ is defined as

$$J^{\uparrow}u = \sum_R u_R \cdot \mathbb{1}_R;$$

where u_R is a scalar value (the component entry of u at $R \in \underline{\mathcal{G}}$) and the sum is taken over all connected components in G_{high}.

As proved in (Koke, 2024), we then have the following:

Theorem I.3. We have

$$\left\|R_z(\Delta) - J^{\uparrow} R_z(\underline{\Delta}) J^{\downarrow}\right\| = \mathcal{O}\left(\frac{\|\Delta_{\text{reg.}}\|}{\lambda_1(\Delta_{\text{high}})}\right)$$

holds; with $\lambda_1(\Delta_{\text{high}})$ denoting the first non-zero eigenvalue of Δ_{high}.

$$\lambda_{\max}(\Delta_{\text{reg.}}) = \|\Delta_{\text{reg.}}\|.$$

We here restate the proof for convenience.

Proof. We will split the proof of this result into multiple steps. For $z < 0$ Let us denote by

$$\begin{aligned} R_z(\Delta) &= (\Delta - zId)^{-1}, \\ R_z(\Delta_{\textit{high}}) &= (\Delta_{\textit{high}} - zId)^{-1} \\ R_z(\Delta_{\textit{reg.}}) &= (\Delta_{\textit{reg.}} - zId)^{-1} \end{aligned}$$

the resolvents correspodning to Δ, $\Delta_{\textit{high}}$ and $\Delta_{\textit{reg.}}$ respectively.
Our first goal is establishing that we may write

$$R_z(\Delta) = [Id + R_z(\Delta_{\textit{high}})\Delta_{\textit{reg.}}]^{-1} \cdot R_z(\Delta_{\textit{high}})$$

This will follow as a consequence of what is called the second resolvent formula Teschl (2014):

"Given self-adjoint operators A, B, we may write

$$R_z(A + B) - R_z(A) = -R_z(A)BR_z(A + B).\text{”}$$

In our case, this translates to

$$R_z(\Delta) - R_z(\Delta_{\textit{high}}) = -R_z(\Delta_{\textit{high}})\Delta_{\text{reg.}}R_z(\Delta)$$

or equivalently

$$[Id + R_z(\Delta_{\textit{high}})\Delta_{\text{reg.}}]\, R_z(\Delta) = R_z(\Delta_{\textit{high}}).$$

Multiplying with $[Id + R_z(\Delta_{\textit{high}})\Delta_{\text{reg.}}]^{-1}$ from the left then yields

$$R_z(\Delta) = [Id + R_z(\Delta_{\textit{high}})\Delta_{\textit{reg.}}]^{-1} \cdot R_z(\Delta_{\textit{high}})$$

as desired.
Hence we need to establish that $[Id + R_z(\Delta_{\textit{high}})\Delta_{\textit{reg.}}]$ is invertible for $z < 0$.

To establish a contradiction, assume it is not invertible. Then there is a signal x such that

$$[Id + R_z(\Delta_{\textit{high}})\Delta_{\textit{reg.}}]\, x = 0.$$

Multiplying with $(\Delta_{\text{high}} - zId)$ from the left yields

$$(\Delta_{\text{high}} + \Delta_{\text{reg.}} - zId)x = 0$$

which is precisely to say that

$$(\Delta - zId)x = 0$$

But since Δ is a graph Laplacian, it only has non-negative eigenvalues. Hence we have reached our contradiction and established

$$R_z(\Delta) = [Id + R_z(\Delta_{\textit{high}})\Delta_{\textit{reg.}}]^{-1}\, R_z(\Delta_{\textit{high}}).$$

Our next step is to establish that

$$R_z(\Delta_{\textit{high}}) \rightarrow \frac{P_0^{\text{high}}}{-z},$$

where P_0^{high} is the spectral projection onto the eigenspace corresponding to the lowest lying eigenvalue $\lambda_0(\Delta_{high}) = 0$ of Δ_{high}. Indeed, by the spectral theorem for finite dimensional operators (c.f. e.g. Teschl (2014)), we may write

$$R_z(\Delta_{high}) \equiv (\Delta_{high} - zId)^{-1} = \sum_{\lambda \in \sigma(\Delta_{high})} \frac{1}{\lambda - z} \cdot P_\lambda^{high}.$$

Here $\sigma(\Delta_{high})$ denotes the spectrum (i.e. the collection of eigenvalues) of Δ_{high} and the $\{P_\lambda^{high}\}_{\lambda \in \sigma(\Delta_{high})}$ are the corresponding (orthogonal) eigenprojections onto the eigenspaces of the respective eigenvalues. Thus we find

$$\left\| R_z(\Delta_{high}) - \frac{P_0^{high}}{-z} \right\| = \left| \sum_{0 < \lambda \in \sigma(\Delta_{high})} \frac{1}{\lambda - z} \cdot P_\lambda^{high} \right\|;$$

where the sum on the right hand side now excludes the eigenvalue $\lambda = 0$.

Using orthonormality of the spectral projections, the fact that $z < 0$ and monotonicity of $1/(\cdot + |z|)$ we find

$$\left\| R_z(\Delta_{high}) - \frac{P_0^{high}}{-z} \right\| = \frac{1}{\lambda_1(\Delta_{high}) + |z|}.$$

Here $\lambda_1(\Delta_{high})$ is the firt non-zero eigenvalue of (Δ_{high}).
Non-zero eigenvalues scale linearly with the weight scale since we have

$$\lambda(S \cdot \Delta) = S \cdot \lambda(\Delta)$$

for any graph Laplacian (in fact any matrix) Δ with eigenvalue λ. Thus we have

$$\left\| R_z(\Delta_{high}) - \frac{P_0^{high}}{-z} \right\| = \frac{1}{\lambda_1(\Delta_{high}) + |z|} \leqslant \frac{1}{\lambda_1(\Delta_{high})} \longrightarrow 0$$

as $\lambda_1(\Delta_{high}) \to \infty$.

Our next task is to use this result in order to bound the difference

$$I := \left\| \left[Id + \frac{P_0^{high}}{-z} \Delta_{reg.} \right]^{-1} \frac{P_0^{high}}{-z} - \left[Id + R_z(\Delta_{high})\Delta_{reg.} \right]^{-1} R_z(\Delta_{high}) \right\|.$$

To this end we first note that the relation

$$[A + B - zId]^{-1} = [Id + R_z(A)B]^{-1} R_z(A)$$

provided to us by the second resolvent formula, implies

$$[Id + R_z(A)B]^{-1} = Id - B[A + B - zId]^{-1}.$$

Thus we have

$$\begin{aligned} \left\| [Id + R_z(\Delta_{high})\Delta_{reg.}]^{-1} \right\| &\leqslant 1 + \|\Delta_{\text{reg.}}\| \cdot \|R_z(\Delta)\| \\ &\leqslant 1 + \frac{\|\Delta_{\text{reg.}}\|}{|z|}. \end{aligned}$$

With this, we have

$$
\begin{aligned}
&\left\|\left[Id + \frac{P_0^{\mathit{high}}}{-z}\Delta_{\mathit{reg.}}\right]^{-1} \cdot \frac{P_0^{\mathit{high}}}{-z} - R_z(\Delta)\right\| \\
=&\left\|\left[Id + \frac{P_0^{\mathit{high}}}{-z}\Delta_{\mathit{reg.}}\right]^{-1} \cdot \frac{P_0^{\mathit{high}}}{-z} - [Id + R_z(\Delta_{\mathit{high}})\Delta_{\mathit{reg.}}]^{-1} \cdot R_z(\Delta_{\mathit{high}})\right\| \\
\leqslant&\left\|\frac{P_0^{\mathit{high}}}{-z}\right\| \cdot \left\|\left[Id + \frac{P_0^{\mathit{high}}}{-z}\Delta_{\mathit{reg.}}\right]^{-1} - [Id + R_z(\Delta_{\mathit{high}})\Delta_{\mathit{reg.}}]^{-1}\right\| + \left\|\frac{P_0^{\mathit{high}}}{-z} - R_z(\Delta_{\text{high}})\right\| \cdot \left\|[Id + R_z(\Delta_{\mathit{high}})\Delta_{\mathit{reg.}}]^{-1}\right\| \\
\leqslant&\frac{1}{|z|}\left\|\left[Id + \frac{P_0^{\mathit{high}}}{-z}\Delta_{\mathit{reg.}}\right]^{-1} - [Id + R_z(\Delta_{\mathit{high}})\Delta_{\mathit{reg.}}]^{-1}\right\| + \left(1 + \frac{\|\Delta_{\text{reg.}}\|}{|z|}\right) \cdot \frac{1}{\lambda_1(\Delta_{\mathit{high}})}.
\end{aligned}
$$

Hence it remains to bound the left hand summand. For this we use the following fact (c.f. Horn & Johnson (2012), Section 5.8. "Condition numbers: inverses and linear systems"):

Given square matrices A, B, C with $C = B - A$ and $\|A^{-1}C\| < 1$, we have

$$\|A^{-1} - B^{-1}\| \leqslant \frac{\|A^{-1}\| \cdot \|A^{-1}C\|}{1 - \|A^{-1}C\|}.$$

In our case, this yields (together with $\|P_0^{\mathit{high}}\| = 1$) that

$$
\begin{aligned}
&\left\|\left[Id + P_0^{\mathit{high}}/(-z) \cdot \Delta_{\mathit{reg.}}\right]^{-1} - [Id + R_z(\Delta_{\mathit{high}})\Delta_{\mathit{reg.}}]^{-1}\right\| \\
\leqslant&\frac{(1 + \|\Delta_{\text{reg.}}\|/|z|)^2 \cdot \|\Delta_{\text{reg.}}\| \cdot \|\frac{P_0^{\text{high}}}{-z} - R_z(\Delta_{\text{high}})\|}{1 - (1 + \|\Delta_{\text{reg.}}\|/|z|) \cdot \|\Delta_{\text{reg.}}\| \cdot \|\frac{P_0^{\text{high}}}{-z} - R_z(\Delta_{\text{high}})\|}
\end{aligned}
$$

For S_{high} sufficiently large, we have

$$\| - P_0^{\text{high}}/z - R_z(\Delta_{\text{high}})\| \leqslant \frac{1}{2\,(1 + \|\Delta_{\text{reg.}}\|/|z|)}$$

so that we may estimate

$$
\begin{aligned}
&\left\|\left[Id + \Delta_{\mathit{reg.}}\frac{P_0^{\mathit{high}}}{-z}\right]^{-1} - [Id + \Delta_{\mathit{reg.}}R_z(\Delta_{\mathit{high}})]^{-1}\right\| \\
\leqslant&2 \cdot (1 + \|\Delta_{\text{reg.}}\|) \cdot \|\frac{P_0^{\text{high}}}{-z} - R_z(\Delta_{\text{high}})\| \\
=&2\frac{1 + \|\Delta_{\text{reg.}}\|/|z|}{\lambda_1(\Delta_{\text{high}})}
\end{aligned}
$$

Thus we have now established

$$\left|\left[Id + \frac{P_0^{\mathit{high}}}{-z}\Delta_{\mathit{reg.}}\right]^{-1} \cdot \frac{P_0^{\mathit{high}}}{-z} - R_z(\Delta)\right| = \mathcal{O}\left(\frac{\|\Delta_{\text{reg.}}\|}{\lambda_1(\Delta_{\text{high}})}\right).$$

Hence we are done with the proof, as soon as we can establish

$$\left[-zId + P_0^{\mathit{high}}\Delta_{\mathit{reg.}}\right]^{-1} P_0^{\mathit{high}} = J^{\uparrow} R_z(\underline{\Delta}) J^{\downarrow},$$

with $J^{\uparrow}, \underline{\Delta}, J^{\downarrow}$ as defined above. To this end, we first note that

$$J^{\uparrow} \cdot J^{\downarrow} = P_0^{\mathit{high}} \tag{6}$$

and

$$J^{\downarrow} \cdot J^{\uparrow} = Id_{\underline{G}}. \tag{7}$$

Indeed,the relation (6) follows from the fact that the eigenspace corresponding to the eignvalue zero is spanned by the vectors $\{\mathbb{1}_R\}_R$, with $\{R\}$ the connected components of G_{high}. Equation (7) follows from the fact that

$$\langle \mathbb{1}_R, \mathbb{1}_R \rangle = \underline{\mu}_R.$$

With this we have

$$\left[Id + P_0^{\textit{high}} \Delta_{\textit{reg.}}\right]^{-1} P_0^{\textit{high}} = \left[Id + J^{\uparrow} J^{\downarrow} \Delta_{\textit{reg.}}\right]^{-1} J^{\uparrow} J^{\downarrow}.$$

To proceed, set

$$\underline{x} := F^{\downarrow} x$$

and

$$\mathscr{X} = \left[P_0^{\textit{high}} \Delta_{\textit{reg.}} - zId\right]^{-1} P_0^{\textit{high}} x.$$

Then

$$\left[P_0^{\textit{high}} \Delta_{\textit{reg.}} - zId\right] \mathscr{X} = P_0^{\textit{high}} x$$

and hence $\mathscr{X} \in \text{Ran}(P_0^{\textit{high}})$. Thus we have

$$J^{\uparrow} J^{\downarrow} (\Delta_{\text{reg.}} - zId) J^{\uparrow} J^{\downarrow} \mathscr{X} = J^{\uparrow} J^{\downarrow} x.$$

Multiplying with $J^{\downarrow}$ from the left yields

$$J^{\downarrow} (\Delta_{\text{reg.}} - zId) J^{\uparrow} J^{\downarrow} \mathscr{X} = J^{\downarrow} x.$$

Thus we have

$$(J^{\downarrow} \Delta_{\text{reg.}} J^{\uparrow} - zId) J^{\uparrow} J^{\downarrow} \mathscr{X} = J^{\downarrow} x.$$

This – in turn – implies

$$J^{\uparrow} J^{\downarrow} \mathscr{X} = \left[J^{\downarrow} \Delta_{\text{reg.}} J^{\uparrow} - zId\right]^{-1} J^{\downarrow} x.$$

Using

$$P_0^{\textit{high}} \mathscr{X} = \mathscr{X},$$

we then have

$$\mathscr{X} = J^{\uparrow} \left[J^{\downarrow} \Delta_{\text{reg.}} J^{\uparrow} - zId\right]^{-1} J^{\downarrow} x.$$

We have thus concluded the proof if we can prove that $J^{\downarrow} \Delta_{\text{reg.}} J^{\uparrow}$ is the Laplacian corresponding to the graph $\underline{G}$ defined in Definition I.1. But this is a straightforward calculation. □

As a corollary, we find

Corollary I.4. *We have*

$$R_z(\Delta)^k \rightarrow J^{\uparrow} R^k(\underline{\Delta}) J^{\downarrow}$$

Proof. This follows directly from the fact that

$$J^{\downarrow} J^{\uparrow} = Id_{\underline{G}}.$$

□

I.2 Further Discussion of Graphs discretizing an Ambient Spaces

Here we further discuss the setting of two graphs discretizing the same ambient space $\mathcal{M}$ in the sense of

$$\|J_i^{\uparrow} e^{-t\Delta_i} J_i^{\downarrow} - e^{-t\Delta_{\mathcal{M}}}\| \leqslant \delta.$$

We will assume $J_i^{\downarrow} J_i^{\uparrow} = Id_{G_i}$, which is a justified assumption, as Example I.5 below elucidates. In this setting, we then have

$$\begin{aligned}
&\|e^{-t\Delta_1} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-t\Delta_2} (J_2^{\downarrow} J_1^{\uparrow})\| \\
=&\|e^{-t\Delta_1} - J_1^{\downarrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\uparrow} + J_1^{\downarrow} (\Delta_{\mathcal{M}} + Id)^{-1} J_1^{\uparrow} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-t\Delta_2} (J_2^{\downarrow} J_1^{\uparrow})\| \\
\leqslant&\|e^{-t\Delta_1} - J_1^{\downarrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\uparrow}\| + \|J_1^{\downarrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\uparrow} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-t\Delta_2} (J_2^{\downarrow} J_1^{\uparrow})\|
\end{aligned}$$

We note

$$\begin{aligned}
&\|e^{-t\Delta_1} - J_1^{\downarrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\uparrow}\| \\
=&\|J_1^{\downarrow} J_1^{\uparrow} e^{-t\Delta_1} J_1^{\downarrow} J_1^{\uparrow} - J_1^{\downarrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\uparrow}\| \\
\leqslant&\|J_1^{\downarrow}\| \|J_1^{\uparrow}\| \cdot \|e^{-t\Delta_1} - J_1^{\uparrow} e^{-t\Delta_{\mathcal{M}}} J_1^{\downarrow}\| \lesssim \delta.
\end{aligned}$$

We consider:

$$\begin{aligned}
&\|e^{-t\Delta_{\mathcal{M}}} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-t\Delta_2} (J_2^{\downarrow} J_1^{\uparrow})\| \\
\leqslant&\|J_1^{\downarrow}\| \|J_1^{\uparrow}\| \cdot \|e^{-t\Delta_{\mathcal{M}}} - J_2^{\uparrow} e^{-t\Delta_2} J_2^{\downarrow}\| \\
\lesssim&\|e^{-t\Delta_{\mathcal{M}}} - J_2^{\uparrow} e^{-t\Delta_2} J_2^{\downarrow}\| \leqslant \delta.
\end{aligned}$$

Hence we have indeed established

$$\|e^{-t\Delta_1} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-t\Delta_2} (J_2^{\downarrow} J_1^{\uparrow})\| \lesssim 2\delta.$$

Next let us consider an explicit example.

Example I.5. To this end, let us revisit the torus-setting introduced in Fig. 13.

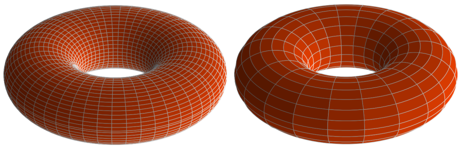

Figure 16: Distinct Torus Discretizations

We begin by recalling that the standard torus $\mathbb{T}$ arises as the cartesian product of two circles S_1 of circumference 2π:

$$\mathbb{T} = S^1 \times S^1.$$

Let us parametrize these circles via angles $0 \leqslant \theta_1, \theta_1 \leqslant 2\pi$. The Laplacian on $\mathbb{T}$ can then be written as

$$\Delta_{\mathbb{T}} = -\partial_{\theta_1}^2 - \partial_{\theta_2}^2.$$

A set of corresponding normalized eigenfunctions are given as

$$\phi_{k_1,k_2} = \frac{1}{2\pi} e^{-ik_1\theta_1} e^{-ik_2\theta_2}$$

with corresponding eigenvalues

$$\lambda_{k_1,k_2} = k_1^2 + k_2^2$$

and $k_1, k_2 \in \mathbb{Z}$.

We now consider a regular discretization of $\mathbb{T}$ using N^2 nodes. This mesh can be thought of as arising from regular discretizations of each S^1 factor; with a node being placed at angles $\phi = \frac{2\pi}{N}k$ with $0 \leqslant k \leqslant N$. The individual node weight of each node in the mesh discretization of $\mathbb{T}$ is set to $\mu = \frac{(2\pi)^2}{N^2}$. We might think of this discretization $\mathbb{T}_N$ pf $\mathbb{T}$ as arising via a cartesian product of the group $\mathbb{Z}/N\mathbb{Z}$ (i.e. the group of integers modulo N) with itself. Each node of $\mathbb{T}_N = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ is then specified by a tuple $(a, b) \in \mathbb{T}_N$, with $a \in \mathbb{Z}/N\mathbb{Z}$ and $b \in \mathbb{Z}/N\mathbb{Z}$.

The graph Laplacian Δ_N on $\mathbb{T}_N = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ then acts on a scalar node signal x_{ab} as

$$(\Delta_N x)_{ab} = \frac{N^2}{(2\pi)^2} \left(4x_{ab} - x_{(a+1)b} - x_{(a-1)b} - x_{a(b+1)} - x_{a(b-1)}\right).$$

Henceforth we will adopt the notation $x(a, b) \equiv x_{ab}$.
Normalized eigenvectors for this Laplacian Δ_N on $\mathbb{T}_N$ are given as

$$\phi^N_{k_1,k_2} = \frac{1}{2\pi} e^{-i\frac{2\pi k_1}{N}a} e^{-i\frac{2\pi k_1}{N}b}$$

with $0 \leqslant k_1, k_2 \leqslant (N-1)$. Corresponding eigenvalues are found to be

$$\lambda^N_{k_1,k_2} = \frac{N^2}{\pi^2}\left[\sin^2\left(\frac{\pi}{N}\cdot k_1\right) + \sin^2\left(\frac{\pi}{N}\cdot k_2\right)\right].$$

To facilitate contact between $\mathbb{T}$ and its graph approximation $\mathbb{T}_N$, we define an interpolation operator $J_N^\uparrow$ that maps a graph signal $f(a, b)$ defined on $\mathbb{T} = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ to a function $\overline{f}$ defined on $\mathbb{T}$ by defining

$$\overline{f}(\theta_1, \theta_2) = f(a, b)$$

whenever $\frac{2\pi}{N}(a-1) \leqslant \theta_1 \leqslant \frac{2\pi}{N}a$ and $\frac{2\pi}{N}(b-1) \leqslant \theta_2 \leqslant \frac{2\pi}{N}b$.
We then take $J^\downarrow$ to be the adjoint of $J^\uparrow$ (i.e. $J^\downarrow = (J^\uparrow)^*$. It is not hard to see that $J^\downarrow J^\uparrow = Id_{\mathbb{T}_N}$.
We now want to show that (for $t > 0$)

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-t\Delta_N} J^\downarrow\| \to 0 \tag{8}$$

as $N \to \infty$. To this end, denote by P_{k_1,K_2} the orthogonal projection onto ϕ_{k_1,k_2}. Denote by $P^N_{k_1,K_2}$ the orthogonal projection onto $\overline{\phi^N_{k_1,k_2}}$. We note

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-t\Delta_N} J^\downarrow\| = \left\| \sum_{k_1,k_2 \in \mathbb{Z}} e^{-\lambda_{k_1,k_2}t} P_{k_1,k_2} - \sum_{--\frac{N-1}{2} \leqslant p_1,p_2 \leqslant \frac{N-1}{2}} e^{-\lambda_{k_1,k_2}t} P^N_{p_1,p_2} \right\|.$$

From this we observe

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-t\Delta_N} J^\downarrow\| = \left\| \sum_{k_1,k_2 \in \mathbb{Z}} e^{-\lambda_{k_1,k_2}t} P_{k_1,k_2} - \sum_{--\frac{N-1}{2} \leqslant p_1,p_2 \leqslant \frac{N-1}{2}} e^{-\lambda^N_{p_1,p_2}t} P^N_{p_1,p_2} \right\|$$

$$\leqslant \left\| \sum_{\frac{N-1}{2} < |k_1|,|k_2|} e^{-\lambda_{k_1,k_2}t} P_{k_1,k_2} \right\| + \left\| \sum_{--\frac{N-1}{2} \leqslant k_1,k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2}t} P_{k_1,k_2} - e^{-\lambda^N_{k_1,k_2}t} P^N_{k_1,k_2} \right) \right\|$$

For the first summand, we already have

$$\left\| \sum_{\frac{N-1}{2} < |k_1|,|k_2|} e^{-\lambda_{k_1,k_2}t} P_{k_1,k_2} \right\| \leqslant e^{-t\frac{(N-1)^2}{2}}.$$

Hence let us investigate the second summand. We note

$$\left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2} t} P_{k_1,k_2} - e^{-\lambda^N_{k_1,k_2} t} P^N_{k_1,k_2} \right) \right\| \tag{9}$$

$$\leqslant \left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2} t} - e^{-\lambda^N_{k_1,k_2} t} \right) P^N_{k_1,k_2} \right\| + \left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} e^{-\lambda_{k_1,k_2} t} (P_{k_1,k_2} - P^N_{k_1,k_2}) \right\|$$

For the first summand we note

$$\begin{aligned} &\left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2} t} - e^{-\lambda^N_{k_1,k_2} t} \right) P^N_{k_1,k_2} \right\| \\ = &\sup_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left| e^{-\lambda_{k_1,k_2} t} - e^{-\lambda^N_{k_1,k_2} t} \right| \\ = &\sup_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} e^{-t(k_1^2+k_2^2)} \left| 1 - e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_1\right) - k_1^2\right)} e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_2\right) - k_2^2\right)} \right| \end{aligned}$$

We note

$$\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k\right) - k^2 \right) = \mathcal{O}\left(\frac{k^4}{N^2}\right).$$

Using

$$\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} N^{\frac{1}{3}}\right) \lesssim N^{\frac{2}{3}}$$

we note

$$\begin{aligned} &\sup_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} e^{-t(k_1^2+k_2^2)} \left| 1 - e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_1\right) - k_1^2\right)} e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_2\right) - k_2^2\right)} \right| \\ \leqslant &\sup_{|k_1|, |k_2| \leqslant N^{\frac{1}{3}}} e^{-t(k_1^2+k_2^2)} \left| 1 - e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_1\right) - k_1^2\right)} e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_2\right) - k_2^2\right)} \right| \\ + &\sup_{|k_1|, |k_2| > N^{\frac{1}{3}}} e^{-t(k_1^2+k_2^2)} \left| 1 - e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_1\right) - k_1^2\right)} e^{-t\left(\frac{N^2}{\pi^2} \sin^2\left(\frac{\pi}{N} k_2\right) - k_2^2\right)} \right| \\ &\leqslant e^{-t(2N^{\frac{2}{3}})} + e^{-t(2N^{\frac{2}{3}})} + e^{-t(N^{\frac{2}{3}})}. \end{aligned}$$

Hence it remains to bound the second summand in (9). We note

$$\begin{aligned} &\left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} e^{-\lambda_{k_1,k_2} t} (P_{k_1,k_2} - P^N_{k_1,k_2}) \right\| \\ \leqslant &\sum_{|k_1|, |k_2| \leqslant \frac{N-1}{2}} e^{-(k_1^2+k_2^2)t} \| P_{k_1,k_2} - P^N_{k_1,k_2} \|. \end{aligned}$$

Next we note

$$\| P_{k_1,k_2} - P^N_{k_1,k_2} \| \leqslant 2 \left\| \phi_{k_1,k_2} - \phi_{k_1,k_2} \right\|.$$

It is not hard to see that

$$\left\| \phi_{k_1,k_2} - \overline{\phi^N_{k_1,k_2}} \right\| \leqslant 2C(|k_1| + |k|_2) \frac{2\pi}{N}$$

for some appropriately chosen $C > 0$. Hence we have

$$\begin{aligned}&\left\|\sum_{-\frac{N-1}{2}\leqslant k_1,k_2\leqslant\frac{N-1}{2}} e^{-\lambda_{k_1,k_2}t}(P_{k_1,k_2}-P^N_{k_1,k_2})\right\|\\ \leqslant&\sum_{|k_1|,|k_2|\leqslant\frac{N-1}{2}} e^{-(k_1^2+k_2^2)t}\cdot 2C(|k_1|+|k|_2)\frac{2\pi}{N}\\ =&\mathcal{O}(1/N).\end{aligned}$$

Where the lass claim follows from summability in k_1, k_2. Thus we have in total indeed established that (8) holds.

I.3 Coarse graining weighted directed Graphs

In this section, following (Koke, 2024) we consider a graph G with directed weighted adjacency matrix A^s which we (disjointly) decompose as

$$A^s \equiv A^c + s\cdot A^m$$

into a weighted directed (partial) adjacency matrix A_C which we keep constant and a weighted directed (partial) adjacency matrix $s\cdot A^m$. Both adjacency matrices determine directed graph structures on the same common node set $\mathcal{G}$. Similar to the setting of Appendix I.1, we are then interested in establishing that when $s\to\infty$ this graph is similar (from a diffusion perspective) to a coarse grained graph $\underline{G}$. In Appendix I.1, we saw that the the coarse grained "limit graph" $\underline{G}$ was determined by the structure of the kernel of the operator Δ_{high}; which encoded the connected components of the graph G_{high} (c.f. Fig. 7) into its vectors. We expect that this also persists in the directed setting.

In this directed setting, we are faced with the choice of whether to make use of the in-degree Laplacian

$$L^{\text{in}} = M^{-1}\left[D^{\text{in}}-A\right]$$

or the out-degree Laplacian

$$L^{\text{out}} = M^{-1}\left[D^{\text{out}}-A\right].$$

The following is known about the kernels of these operators (c.f. Veerman & Lyons (2020); Sahi (2013)):

In-degree Laplacian: To understand the kernel of directed in-degree Laplacians, we need the concept of reaches. Reaches generalize the concept of connected components of undirected graphs Veerman & Lyons (2020): A subgraph $R\subseteq G$ is called reach, if for any two vertices $a,b\in R$ there is a directed path in R along which the (directed) edge weights do not vanish, and R simultaneously possesses no outgoing connections (i.e. for any $c\in G$ with $c\notin R$: $w_{ca}=0$). We here limit ourselves to the setting where all reaches within a given graph are disjoint (c.f. Veerman & Lyons (2020) for the general setting).

Consider now a graph G with adjacency matrix A^m The dimensionality of the kernel of L^{in} on this graph is then given as the number of reaches N_{Reach} present in A^m. The right-kernel of L^{in} is spanned by the vectors $\{v_i\}_{1\leqslant R\leqslant N_{\text{Reach}}}$ which have entry 1 at all nodes in reach R and are zero outside of R. By definition these vectors satisfy

$$L^{\text{in}}\cdot v_i = 0.$$

The left-kernel is spanned by vectors $\{w_R\}_{1\leqslant R\leqslant N_{\text{Reach}}}$ so that w_R has non-zero entries only for nodes in reach R and is zero elsewhere. As can be derived from results in Sahi (2013), we may write $w_R = M\hat{w}_R$ with M the matrix of node weights (c.f. Section 2.1) and the entry $(\hat{w}_R)_i$ (for i a node in the reach R) given as

$$(\hat{w}_R)_i = \sum_{\tau_i\in\mathcal{T}_i^R}\prod_{(ab)\in\tau_i} A^m_{ab}.$$

Here $\mathcal{T}_i^R$ is the set of all spanning trees of the reach R that are rooted at node $i\in R$. τ_i is such a spanning tree beginning at node i. The quantity $\prod\limits_{(ab)\in\tau_i} A^m_{ab}$ then multiplies all (directed) edge weights along the spanning tree τ_i. From this, we can derive that we may write the (not necessarily orthogonal) projection P projecting onto the kernel of L^{in} as

$$P = \sum_{R \in \text{Reaches of } A^m} \frac{v_R \cdot (M\hat{w}_R)^\intercal}{(M\hat{w}_R)^\intercal \cdot V_R}.$$

We might write this as

$$P = J^\uparrow J^\downarrow$$

with $J^\downarrow$ mapping (similarly to the setting in Appendix I.1) to a coarsified graph $\underline{G}$, whose node set consists of the reaches in the original graph structure determined by A:

$$\underline{\mathcal{G}} = \{R\}_{R \in \{\text{Reaches of } A^m\}}.$$

Similarly to Definition I.2, we then have for x a signal defined on the original graph G, that $(J^\downarrow x)$ is a signal on the coarsified graph $\underline{G}$. It is defined by specifying it on each node $R \in \underline{\mathcal{G}}$ as

$$(J^\downarrow x)_R = \frac{1}{(M\hat{w}_R)^\intercal \cdot V_R} \cdot (M\hat{w}_R)^\intercal \cdot x.$$

Similarly interpolation back up to G is defined as

$$J^\uparrow \underline{x} := \sum_{R \in \underline{\mathcal{G}}} \underline{x}_R \cdot v_R.$$

Out-degree Laplacian: For the out-degree Laplacian L^{out}, the roles of left- and right kernels above are essentially reversed. Instead of reaches R determined by the adjacency matrix A^m, one considers reaches $\tilde{R}$ determined by the transpose $(A^m)^\intercal$ of the adjacency matrix. The left kernel of the out-degree Laplacian is given as the set of vectors $\{\tilde{v}_{\tilde{R}}\}$ given as $\tilde{v}_{\tilde{R}} = Mv_{\tilde{R}}$, with

$v_{\tilde{R}}$ again the vector with entry 1 at all nodes in reach $\tilde{R}$ and zero outside of $\tilde{R}$. The right kernel is spanned by vectors $\{\tilde{w}_{\tilde{R}}\}$ whose ith entry is given by

$$(\tilde{w}_{\tilde{R}})_i = \sum_{\tilde{\tau}_i \in \mathcal{T}_i^{\tilde{R}}} \prod_{(ab) \in \tilde{\tau}_i} A^\intercal_{ab}.$$

Here $\mathcal{T}_i^{\tilde{R}}$ is the set of all spanning trees of the reach $\tilde{R}$ (as determined by the connectivity structure of the transposed adjacency matrix $(A^m)^\intercal$).

We then note for the projection $\tilde{P}$ onto the kernel of L^{out}, that we may write

$$\tilde{P} = \sum_{\tilde{R} \in \text{Reaches of } (A^m)^\intercal} \frac{\tilde{w}_{\tilde{R}} \cdot (Mv_{\tilde{R}})^\intercal}{(Mv_{\tilde{R}})^\intercal \cdot \tilde{w}_{\tilde{R}}}.$$

We may again write this as

$$P = \tilde{J}^\uparrow \tilde{J}^\downarrow$$

with $J^\downarrow$ mapping (similarly to the setting in Appendix I.1) to a coarsified graph $\underline{G}$, whose node set consists of the reaches in the adjacency structure determined by $(A^m)^\intercal$:

Similarly to above, we then have for x a signal defined on the original graph G, that $(\tilde{J}^\downarrow x)$ is a signal on the coarsified graph $\underline{G}$. It is defined by specifying it on each node $\tilde{R} \in \underline{\mathcal{G}}$ as

$$(\tilde{J}^\downarrow x)_{\tilde{R}} = \frac{1}{(Mv_{\tilde{R}})^\intercal \cdot \tilde{w}_{\tilde{R}}} \cdot (Mv_{\tilde{R}})^\intercal \cdot x$$

Similarly interpolation back up to G is defined as

$$\tilde{J}^\uparrow \underline{x} := \sum_{\tilde{R} \in \underline{\mathcal{G}}} \underline{x}_R \cdot \tilde{w}_{\tilde{R}}.$$

In the setting

$$A_s \equiv A_c + s \cdot A^m$$

we may then prove (exactly as done in Appendix I.1) that – with L_s^{in}, L_s^{out} the in-and out-degree Laplacians corresponding to A_s – we have

$$\|(L_s^{\text{in}} + Id)^{-1} - J^{\downarrow}(\underline{L}^{\text{in}} + Id)^{-1} J^{\uparrow}\| = \mathcal{O}\left(\frac{1}{s}\right)$$

and

$$\|(L_s^{\text{out}} + Id)^{-1} - \tilde{J}^{\downarrow}(\underline{L}^{\text{out}} + Id)^{-1} \tilde{J}^{\uparrow}\| = \mathcal{O}\left(\frac{1}{s}\right).$$

Investigating the operators $J^{\uparrow}$ and $\tilde{J}^{\uparrow}$, we see that we have

$$J^{\uparrow}\mathbb{1}_{\underline{G}} = \mathbb{1}_G$$

$$\tilde{J}^{\uparrow}\mathbb{1}_{\underline{G}} \neq \mathbb{1}_G.$$

In view of Theorem H.17 we hence find:

Proposition I.6. *In the directed setting, using the in-degree Laplacian allows for networks to be transferable between a graph G and its coarse grained version* $\underline{G}$ *even if biases are enabled. This is not true when using the out-degree Laplacian.*

J Additional Experimental Considerations

J.1 Additional details on Coarse Graining Examples

Dataset: The dataset we consider is the **QM7** dataset, introduced in Blum & Reymond (2009); Rupp et al. (2012). This dataset contains descriptions of 7165 organic molecules, each with up to seven heavy atoms, with all non-hydrogen atoms being considered heavy. A molecule is represented by its Coulomb matrix C^{Clmb}, whose off-diagonal elements

$$C_{ij}^{\text{Clmb}} = \frac{Z_i Z_j}{|R_i - R_j|}$$

correspond to the Coulomb-repulsion between atoms i and j. We discard diagonal entries of Coulomb matrices; which would encode a polynomial fit of atomic energies to nuclear charge Rupp et al. (2012).

For each atom in any given molecular graph, the individual Cartesian coordinates R_i and the atomic charge Z_i are (in principle) also accessible individually. To each molecule an atomization energy - calculated via density functional theory - is associated. The objective is to predict this quantity. The performance metric is mean absolute error. Numerically, atomization energies are negative numbers in the range -600 to -2200. The associated unit is [*kcal/mol*].

Details on collapsing procedure: Again, we make use of the QM7 dataset Rupp et al. (2012) and its Coulomb matrix description

$$C_{ij}^{\text{Clmb}} = \frac{Z_i Z_j}{|R_i - R_j|} \tag{10}$$

of molecules. We modify (all) molecular graphs in QM7 by deflecting hydrogen atoms (H) out of their equilibrium positions towards the respective nearest heavy atom. This is possible since the QM7 dataset also contains the Cartesian coordinates of individual atoms. Edge weights between heavy atoms then remain the same, while Coulomb repulsions between H-atoms and respective nearest heavy atom increasingly diverge; as is evident from (10).

Given an original molecular graph G with node weights $\mu_i = Z_i$, the corresponding limit graph $\underline{G}$ corresponds to a coarse grained description, where heavy atoms and surrounding H-atoms are aggregated into single super-nodes.

Mathematically, $\underline{G}$ is obtained by removing all nodes corresponding to H-atoms from G, while adding the corresponding charges $Z_H = 1$ to the node-weights of the respective nearest heavy atom. Charges in (10) are modified similarly to generate the weight matrix $\underline{W}$.

On original molecular graphs, atomic charges are provided via one-hot encodings. For the graph of methane – consisting of one carbon atom with charge $Z_C = 6$ and four hydrogen atoms of charges $Z_H = 1$ – the corresponding node-feature-matrix is e.g. given as

$$X = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 0\cdots \\ 1 & 0 & \cdots & 0 & 0 & 0\cdots \\ 1 & 0 & \cdots & 0 & 0 & 0\cdots \\ 1 & 0 & \cdots & 0 & 0 & 0\cdots \\ 1 & 0 & \cdots & 0 & 0 & 0\cdots \end{pmatrix}$$

with the non-zero entry in the first row being in the 6th column, in order to encode the charge $Z_C = 6$ for carbon.

The feature vector of an aggregated node represents charges of the heavy atom and its neighbouring H-atoms jointly.

Node feature matrices are translated as $\underline{X} = J^{\downarrow} X$. Applying $J^{\downarrow}$ to one-hot encoded atomic charges yields (normalized) bag-of-word embeddings on $\underline{G}$: Individual entries of feature vectors encode how much of the total charge of the super-node is contributed by individual atom-types. In the example of methane, the limit graph $\underline{G}$ consists of a single node with node-weight

$$\mu = 6 + 1 + 1 + 1 + 1 = 10.$$

The feature matrix

$$\underline{X} = J^{\downarrow} X$$

is a single row-vector given as

$$\underline{X} = \left(\frac{4}{10}, 0, \cdots, 0, \frac{6}{10}, 0, \cdots\right).$$

Experimental Setup: We randomly select 1500 molecules for testing and train on the remaining graphs. On QM7 we run experiments for 23 different random random seeds and report mean and standard deviation. All experiments were performed on a single NVIDIA Quadro RTX 8000 graphics card.

Additional details on training and models: Typical GNN models are divided into **standard** architectures (GCN (Kipf & Welling, 2017), ChebNet (Defferrard et al., 2016), ARMA (Bianchi et al., 2019), BernNet (He et al., 2021), GATv2 (Brody et al., 2022)) and **multi- scale** architectures (PushNet (Busch et al., 2020), UFGNet (Zheng et al., 2021), Lanczos (Liao et al., 2019)). Apart from UFGNet (already acting as a **pooling** layer) we also consider self-attention-pooling (Lee et al., 2019); both acting on the final layer (SAG) and as acting on the output of each indivifual layer, with resulting layer-wise features concatenated to produce the final embedding (SAG-M). All considered convolutional layers are incorporated into a two layer deep and fully connected graph convolutional architecture. In each hidden layer, we set the width (i.e. the hidden feature dimension) to

$$F_1 = F_2 = 64.$$

For BernNet, we set the polynomial order to $K = 3$ to combat appearing numerical instabilities. ARMA is set to $K = 2$ and $T = 1$. ChebNet uses $K = 2$. Lnaczos uses 20 Lanczos iterations, as proposed in the original paper (Liao et al., 2019). UFGNet uses Haar wavelets. For all baselines, the standard mean-aggregation scheme is employed after the graph-convolutional layers to generate graph level features. Finally, predictions are generated via an MLP.

LTF-Res architecture, we set $\lambda = 1$ and and build filters using the $k = 1$ and $= 2$ atoms in $\Psi^{\text{Res}} = \{(z + \lambda)^{-k}\}_{k \in \mathbb{N}}$.

For the LTF-Exp architecture, we set $t = 1$ and and build filters using the $k = 1$ and $= 2$ atoms in $\Psi^{\text{Exp}} = \{e^{-(kt_0)z}\}_{k \in \mathbb{N}}$.

As aggregation, we employ the graph level feature aggregation scheme introduced in Section 3.2 with node weights set to atomic charges of individual atoms. Predictions are then generated via a final MLP with the same specifications as the one used for baselines.

J.2 Further discussions on transferability results in Table 1 using Figure 9

Fig. 9 showcases why LTF based models in Table 1 are able to transfer. While it is true that $\lim_{t\to\infty} \eta(t) = 0$, the key take-away here is not that the functions $\eta(t)$ decays to zero, but rather that it decays to zero sufficiently fast. For $t = 1$, we e.g. already have $\eta(1) \approx 0$.

Let us exemplarily examine the implications of this sufficiently fast decay of the function $\eta(t)$ for the transferability of the filter $\psi(z) = e^{-z}$. which constitutes a basis element in our investigated LTF-Exp architecture. The generalized function associated to this filter is given by $\hat{\psi}(t) = \delta(t-1)$.

As discussed in Theorem 4.4 (line 274 ff.) the single filter transferability error is bounded as

$$\|\psi(L) - \tilde{J}\psi(\tilde{L})J\| \leqslant \int_0^\infty \eta(t)|\hat{\psi}(t)|dt = \int_0^\infty \eta(t)\delta(t-1)dt = \eta(1) \approx 0.$$

Since $\eta(1) \approx 0$, the transferability error of the corresponding filter ψ is small. Together with Theorem 4.9 this then explains the transferability observed in Table 1.

J.3 Additional experimental Results on QM9

Here we provide additional experimental results on QM9

Table 3: Regression Mean Absolute Errors (various targets) using high- and low-resolution QM9

	Zero point vibrational energy [eV] (↓)				Dipole Moment [D] (↓)			
Training	**High Resolution**		**Low Resolution**		**High Resolution**		**Low Resolution**	
Inference	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**
GATv2	$3.6464_{\pm 0.0597}$	$0.1785_{\pm 0.0015}$	$0.1328_{\pm 0.0061}$	$5.0610_{\pm 3.3775}$	$3.6551_{\pm 1.7807}$	$0.8816_{\pm 0.0336}$	$0.7851_{\pm 0.017}$	$1.7071_{\pm 0.1063}$
GCN	$0.8463_{\pm 0.0658}$	$0.1851_{\pm 0.0041}$	$0.1344_{\pm 0.0040}$	$0.8243_{\pm 0.0903}$	$2.9901_{\pm 0.4030}$	$0.9237_{\pm 0.0137}$	$0.9594_{\pm 0.0200}$	$1.4992_{\pm 0.1135}$
LTF-Ψ^{Res}	$0.0675_{\pm 0.0115}$	$0.0357_{\pm 0.0062}$	$0.0398_{\pm 0.0022}$	$0.0403_{\pm 0.0026}$	$1.3071_{\pm 0.2227}$	$0.7523_{\pm 0.0094}$	$0.9556_{\pm 0.0263}$	$0.9659_{\pm 0.0202}$
	Free energy at 298.15K [eV] (↓)				Rotational constant [GHz] (↓)			
Training	**High Resolution**		**Low Resolution**		**High Resolution**		**Low Resolution**	
Inference	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**
GATv2	$1252.14_{\pm 787.48}$	$409.44_{\pm 74.09}$	$409.54_{\pm 161.55}$	$2418.55_{\pm 637.45}$	$0.9654_{\pm 0.0480}$	$0.8482_{\pm 0.0674}$	$0.8479_{\pm 0.0223}$	$1.7811_{\pm 0.7105}$
GCN	$11017.24_{\pm 1621.28}$	$344.23_{\pm 15.85}$	$940.03_{\pm 14.38}$	$3588.13_{\pm 366.20}$	$1.4153_{\pm 0.0354}$	$0.7996_{\pm 0.0091}$	$0.8544_{\pm 0.0275}$	$1.0928_{\pm 0.1043}$
LTF-Ψ^{Res}	$18.00_{\pm 5.28}$	$18.00_{\pm 5.28}$	$11.71_{\pm 2.46}$	$11.71_{\pm 2.46}$	$0.9138_{\pm 0.09510}$	$0.8810_{\pm 0.0655}$	$0.8211_{\pm 0.0192}$	$0.9531_{\pm 0.1842}$

J.4 Transferability on Graphs generated via Stochastic Block Models

Stochastic Block Models: Stochastic block models (Holland et al., 1983) are generative models for random graphs that produce graphs containing strongly connected communities. In our experiments in this section, we consider a stochastic block model whose distributions is characterized by four parameters: The number of communities c_{number} determine how many (strongly connected) communities are present in the graph that is to be generated. The community size c_{size} determines the number of nodes belonging to each (strongly connected) community. The probability p_{connect} determines the probability that two nodes within the same community are connected by an edge. The probability p_{inter} determines the probabilities that two nodes in *different* communities are connected by an edge.

Experimental Setup: Since stochastic block models do not generate node-features, we equip each node with a randomly-generated unit-norm feature vector. Given such a graph G drawn from a stochastic block model, we then compute a version $\underline{G}$ of this graph, where all communities are collapsed to single nodes as described in Definition I.2. We then compare the feature vectors generated for G and $\underline{G}$. All experiments were performed on a single NVIDIA Quadro RTX 8000 graphics card.

As before, we then consider the LTF-Ψ^{Res} and LTF-Ψ^{Exp} together with GCN as a baseline when investigating transferability.

Experiment: Varying the Connectivity within the Communities: As discussed in detail in Section 3.2 and Appendix I.1, we desire that networks assign similar feature vectors to graphs with strongly connected communities and coarse-grained versions of these graphs, where these communities are collapsed to aggregate nodes. The higher the connectivity within these communities, the more similar should the feature vector of the original graph G and its coarsified version $\underline{G}$ be, as Appendix I.1 established. In order to verify this experimentally, we fix the parameters $c_{\text{number}}, c_{\text{size}}$ and p_{inter} in our stochastic block model. We then vary the probability p_{connect} that two nodes within the same community are connected by an edge from $p_{\text{connect}} = 0$ to $p_{\text{connect}} = 1$. This corresponds to varying the connectivity within the communities from very sparse (or in fact no connectivity) to full connectivity (i.e. the community being a clique). In Figure 17 below, we then plot the difference of feature vectors generated by LTF-Res, LTF-Exp and GCN for G and $\underline{G}$ respectively. For each $p_{\text{connect}} \in [0, 1]$, results are averaged over 100 graphs randomly drawn from the same stochastic block model.

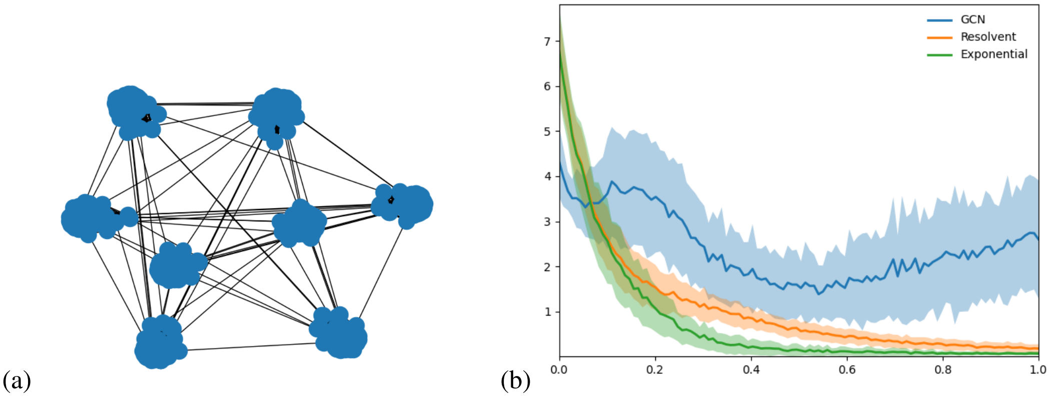

Figure 17: (a) Example Graph (b) Varying the parameter $p_{\text{connect}} \in [0, 1]$ for fixed $c_{\text{size}} = 20$, $p_{\text{inter}} = 2/c^2_{\text{size}}$ and $c_{\text{number}} = 10$.

We have chosen $p_{\text{inter}} = 2/c^2_{\text{size}}$ so that – on average – *clusters* are connected by two edges. The choice of two edges (as opposed to $1, 3, 4, 5, ...$) between clusters is not important; any arbitrary choice of p_{inter} ensures a decay behavior for ResolvNet as in Figure 17. A corresponding ablation study is provided below.

As can be inferred from Fig. 17, LTF-Ψ^{Res} and LTF-Ψ^{Exp} produce more and more similar feature-vectors for G and its coarse-grained version $\underline{G}$, as the connectivity within the clusters is increased. As a reference, we plot GCN for which such a transferability result clearly does not hold.

J.5 Node Level Transferability and Graphs with Varying Connectivity

In the preceding experiments, standard methods proved not transferable. Here we show that this lack of transferability can be harmful also for node-level tasks on a single graph that has an imbalanced geometry in the sense that it contains strongly connected subgraphs with weaker connectivity between such subgraphs.

To this end, we duplicated individual nodes on popular node-classification datasets (Citeseer & Cora (Sen et al., 2008; McCallum et al., 2000)) k-times to form (fully connected) k-cliques, while keeping the train-val-test partition constant.
Models were then trained on the same (k-fold expanded) train-set and asked to classify nodes on the (k-fold expanded) test-partition. Baselines were chosen to form a representative selection of common information-propagation methods and include GIN Xu et al. (2019) and SAGE Hamilton et al. (2017) (which could not handle weighted edges).

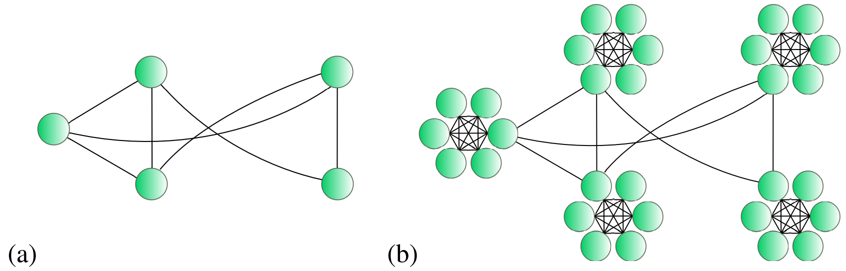

Figure 18: Individual nodes (a) replaced by k-cliques (b)

Additional details on training and models: All experiments were performed on a single NVIDIA Quadro RTX 8000 graphics card. We closely follow the experimental setup of Gasteiger et al. (2019b) on which our codebase builds: All models are trained for a fixed maximum (and unreachably high) number of $n = 10000$ epochs. Early stopping is performed when the validation performance has not improved for 100 epochs. Test-results for the parameter set achieving the highest validation-accuracy are then reported. Ties are broken by selecting the lowest loss (c.f. Velickovic et al. (2018)). Confidence intervals are calculated over multiple splits and random seeds at the 95% confidence level via bootstrapping.

We train all models on a fixed learning rate of lr $= 0.1$. Global dropout probability p of all models is optimized individually over $p \in \{0.3, 0.35, 0.4, 0.45, 0.5\}$. We use ℓ^2 weight decay and optimize the weight decay parameter λ for all models over $\lambda \in \{0.0001, 0.0005\}$. Where applicable (e.g. not for He et al. (2021)) we choose a two-layer deep convolutional architecture with the dimensions of hidden features optimized over

$$K_\ell \in \{32, 64, 128\}. \tag{11}$$

In addition to the hyperparemeters specified above, some baselines have additional hyperparameters, which we detail here: BernNet uses an additional in-layer dropout rate of dp_rate $= 0.5$ and for its filters a polynomial order of $K = 10$ as suggested in He et al. (2021). Hyperparameters depth T and number of stacks K of the ARMA convolutional layer Bianchi et al. (2019) are set to $T = 1$ and $K = 2$. ChebNet also uses $K = 2$ to avoid the known over-fitting issue Kipf & Welling (2017) for higher polynomial orders. The graph attention network Velickovic et al. (2018) uses 8 attention heads, as suggested in Velickovic et al. (2018).

For the LTF-models, we optimize depth over $K = 1, 2$ with hidden feature dimension optimized over the values in (11) as for baselines. We empirically observed in the setting of *unweighted* graphs, that rescaling the Laplacian as

$$\Delta_{nf} := \frac{1}{c_{nf}} \Delta$$

with a normalizing factor c_{nf} on which we base our ResolvNet architectures improved performance.

We express this normalizing factor in terms of the largest singular value $\|\Delta\|$ of the (non-normalized) graph Laplacian. It is then selected among

$$c_{nf}/\|\Delta\| \in \{0.001, 0.01, 0.1, 2\}.$$

The value λ for the resolvent is selected among

$$\lambda \in \{0.14, 0.15, 0.2, 0.25\}.$$

J.6 Transferability between Graphs discretizing a common Ambient Space: The Torus

We make use of the operators $J_i^{\uparrow\downarrow}$ defined in Appendix I.2. The function $f \in L^2(\mathcal{M})$ on the torus is chosen as

$$f = \frac{1}{4\pi^2} \sin(\phi) \cos(\theta).$$

All networks have two hidden layers of width 64 and are asked to predict a scalar signal on the respective graphs.

K Effective Propagation Schemes

For definiteness, we here discuss limit-propagation schemes in the setting where **edge-weights** are large. The discussion for high-connectivity in the Sense of large cliques proceeds analogously.

In this section, we then take up again the setting of Section 3.2. We reformulate this setting here in a slightly modified language, that is more adapted to discussing effective propagation schemes of standard architectures:

We partition edges on a weighted graph G, into two disjoint sets $\mathcal{E} = \mathcal{E}_{\text{reg.}} \dot{\cup} \mathcal{E}_{\text{high}}$, where the set of edges with large weights is given by:

$$\mathcal{E}_{\text{high}} := \{(i,j) \in \mathcal{E} : w_{ij} \geqslant S_{\text{high}}\}$$

and the set with small weights is given by:

$$\mathcal{E}_{\text{reg.}} := \{(i,j) \in \mathcal{E} : w_{ij} \leqslant S_{\text{reg.}}\}$$

for weight scales $S_{\text{high}} > S_{\text{reg.}} > 0$. Without loss of generality, assume $S_{\text{reg.}}$ to be as low as possible (i.e. $S_{\text{reg.}} = \max_{(i,j)\in\mathcal{E}_{\text{reg.}}} w_{ij}$) and S_{high} to be as high as possible (i.e. $S_{\text{large}} = \min_{(i,j)\in\mathcal{E}_{\text{high}}}$) and no weights in between the scales.

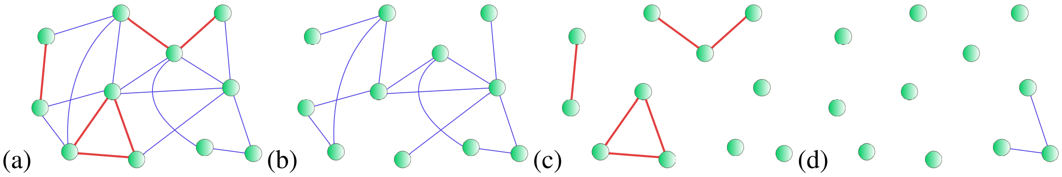

Figure 19: (a) Graph G with $\mathcal{E}_{\text{reg.}}$ (blue) & $\mathcal{E}_{\text{high}}$ (red); (b) $G_{\text{reg.}}$; (c) G_{high}; (d) $G_{\text{reg., exclusive}}$

This decomposition induces two graph structures corresponding to the disjoint edge sets on the node set $\mathcal{G}$: We set $G_{\text{reg.}} := (\mathcal{G}, \mathcal{E}_{\text{reg.}})$ and $G_{\text{high}} := (\mathcal{G}, \mathcal{E}_{\text{high}})$ c.f. Fig. 19).
We also introduce the set of edges $\mathcal{E}_{\text{reg., exclusive}} := \{(i,j) \in \mathcal{E}_{\text{reg.}} | \, \forall k \in \mathcal{G} : \; (i,k) \notin \mathcal{E}_{\text{high}} \;\&\; (k,j) \notin \mathcal{E}_{\text{high}}\}$ connecting nodes that do not have an incident edge in $\mathcal{E}_{\text{high}}$. A corresponding example-graph $G_{\text{reg., exclusive}}$ is depicted in Fig. 19 (d).

We are now interested in the behaviour of graph convolution schemes if the scales are well separated:

$$S_{\text{high}} \gg S_{\text{reg.}}$$

K.1 Spectral Convolutional Filters

We first discuss resulting limit-propagation schemes for spectral convolutional networks. Such networks implement convolutional filters as a mapping

$$x \longmapsto g_{\theta}(T)x$$

for a node feature x, a learnable function g_θ and a graph shift operator T.

K.1.1 Need for Normalization

The graph shift operator T facilitating the graph convolutions needs to be normalized for established spectral graph convolutional architectures:

For Bianchi et al. (2019), this e.g. arises as a necessity for convergence of the proposed implementation scheme for the rational filters introduced there (c.f. eq. (10) in Bianchi et al. (2019)).

The work Defferrard et al. (2016) needs its graph shift operator to be normalized, as it approximates generic filters via a Chebyshev expansion. As argued in Defferrard et al. (2016), such Chebyshev polynomials form an orthogonal basis for the space $L^2([-1,1], dx/\sqrt{1-x^2})$. Hence, the spectrum of the operator T to which the (approximated and learned) function g_θ is applied needs to be contained in the interval $[-1,1]$.

In Kipf & Welling (2017), it has been noted that for the architecture proposed there, choosing T to have eigenvalues in the range $[0,2]$ (as opposed to the normalized ranges $[0,1]$ or $[-1,1]$) has the potential to lead to vanishing- or exploding gradients as well as numerical instabilities. To alleviate this, Kipf & Welling (2017) introduces a "renormalization trick" (c.f. Section 2.2. of Kipf & Welling (2017) to produce a normalized graph shift operator on which the network is then based.

We can understand the relationship between normalization of graph shift operator T and the stability of corresponding convolutional filters explicitly: Assume that we have

$$\|T\| \gg 1.$$

This might e.g. happen when basing networks on the un-normalized graph Laplacian Δ or the weight-matrix W if edge weights are potentially large (such as in the setting $S_{\text{high}} \gg S_{\text{reg.}}$ that we are considering).

By the spectral mapping theorem (see e.g. Teschl (2014)), we have

$$\sigma\left(g_\theta(T)\right) = \left\{g_\theta(\lambda) : \lambda \in \sigma(T)\right\}, \tag{12}$$

with $\sigma(T)$ denoting the spectrum (i.e. the set of eigenvalues) of T. For the largest (in absolute value) eigenvalue $\lambda_{\max}$ of T, we have

$$|\lambda_{\max}| = \|T\|. \tag{13}$$

Since learned functions are either implemented directly as a polynomial (as e.g. in Defferrard et al. (2016); He et al. (2021)) or approximated as a Neumann type power iteration (as e.g. in Bianchi et al. (2019); Gasteiger et al. (2019a)) which can be thought of as a polynomial, we have

$$\lim_{\lambda \to \pm\infty} |g_\theta(\lambda)| = \infty.$$

Thus in view of (12) and (13) we have for $\|T\|$ sufficiently large, that

$$\|g_\theta(T)\| = |g_\theta(\pm\|T\|)|$$

with the sign ± determined by $\lambda_{\max} \gtrless 0$. Since non-constant polynomials behave at least linearly for large inputs, there is a constant $C > 0$ such that

$$C \cdot \|T\| \leqslant \|g_\theta(T)\|$$

for all sufficiently large $\|T\|$. We thus have the estimate

$$\|x\| \cdot C \cdot \|T\| \leqslant \|g_\theta(T)x\|$$

for at least one input signal x (more precisely all x in the eigen-space corresponding to the largest (in absolute value) eigenvalue $\lambda_{\max}$). Thus if T is not normalized (i.e. $\|T\|$ is not sufficiently bounded), the norm of (hidden) features might increase drastically when moving from one (hidden) layer to the next. This behaviour persists for all input signals x have components in eigenspaces corresponding to large (in absolute value) eigenvalues of T.

K.1.2 Spectral Normalizations

As discussed in the previous Section K.1.1, instabilities arising from non-normalized graph shift operators can be traced back to the problem of such operators having large eigenvalues. It was thus – among other considerations – suggested in Defferrard et al. (2016) to base convolutional filters on the spectrally normalized graph shift operator

$$T = \frac{1}{\lambda_{\max}(\Delta)}\Delta,$$

with Δ the un-normalized graph Laplacian. In the setting $S_{\text{high}} \gg S_{\text{reg.}}$ we are considering, this leads to an effective feature propagation along G_{high} (c.f. also Fig. 20) only, as Theorem K.1 below establishes:

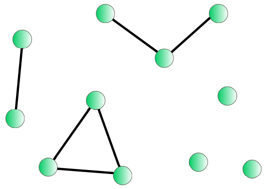

Figure 20: Limit graph corresponding to Fig 19 for spectral normalization

Theorem K.1. With

$$T = \frac{1}{\lambda_{\max}(\Delta)}\Delta,$$

and the scale decomposition as above we have that

$$\left\| T - \frac{1}{\lambda_{\max}(\Delta_{\text{high}})}\Delta_{\text{high}} \right\| = \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right) \tag{14}$$

for $S_{\text{high}} \gg S_{\text{reg.}}$.

Proof. For convenience in notation, let us write

$$T_{\text{high}} = \frac{1}{\lambda_{\max}(\Delta_{\text{high}})}\Delta_{\text{high}}$$

and similarly

$$T_{\text{reg.}} = \frac{1}{\lambda_{\max}(\Delta_{\text{reg.}})}\Delta_{\text{reg.}}.$$

We may write

$$\Delta = \Delta_{\text{high}} + \Delta_{\text{reg.}},$$

which we may rewrite as

$$\Delta = \lambda_{\max}(\Delta_{\text{high}}) \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}}\right). \tag{15}$$

Let us consider the equivalent expression

$$\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta = T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}}. \tag{16}$$

We next note that

$$\lambda_{\max}\left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta\right) = \frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})}. \tag{17}$$

and

$$\lambda_{\max}\left(T_{\text{high}}\right) = 1$$

since the operation of taking eigenvalues of operators is multiplicative in the sense of

$$\lambda_{\max}(|a| \cdot T) = |a| \cdot \lambda_{\max}(T)$$

for non-negative $|a| \geqslant 0$.

Since the right-hand-side of (16) constitutes an analytic perturbation of T_{high}, we may apply analytic perturbation theory (c.f. e.g. Kato (1976) for an extensive discussion) to this problem. With this (together with $\|T_{\text{high}}\| = 1$) we find

$$\lambda_{\max}\left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta\right) = 1 + \mathcal{O}\left(\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})}\right). \tag{18}$$

Using (17) and the fact that

$$\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}}, \tag{19}$$

we thus have

$$\frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})} = 1 + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right).$$

Since for small ϵ, we also have

$$\frac{1}{1+\epsilon} = 1 + \mathcal{O}(\epsilon),$$

the relation (19) also implies

$$\frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}(\Delta)} = 1 + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right).$$

Multiplying (15) with $1/\lambda_{\max}(\Delta)$ yields

$$T = \frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}(\Delta)} \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}} \right). \tag{20}$$

Since $\|T_{\text{high}}\|, \|T_{\text{reg.}}\| = 1$ and

$$\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}} < 1$$

for sufficiently large S_{high}, relation (20) implies

$$\left\| T - \frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \Delta_{\text{high}} \right\| = \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}} \right)$$

as desired.

Note that we might in principle also make use of Lemma K.2 below, to provide quantitative bounds: Lemma K.2 states that

$$|\lambda_k(A) - \lambda_k(B)| \leqslant \|A - B\|$$

for self-adjoint operators A and B and their respective k^{th} eigenvalues ordered by magnitude. On a graph with N nodes, we clearly have $\lambda_{\max} = \lambda_N$ for eigenvalues of (rescaled) graph Laplacians, since all such eigenvalues are non-negative. This implies for the difference $|1 - \lambda_{\max}(\Delta)/\lambda_{\max}(\Delta_{\text{high}})|$ arising in (18) that explicitly

$$\left| 1 - \frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})} \right| \leqslant \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})}.$$

This in turn can then be used to provide a quantitative bound in (14). Since we are only interested in the qualitative behaviour for $S_{\text{high}} \gg S_{\text{reg.}}$, we shall however not pursue this further.

□

It remains to state and establish Lemma K.2 referenced at the end of the proof of Theorem K.1:

Lemma K.2. Let A and B be two hermitian $n \times n$ dimensional matrices. Denote by $\{\lambda_k(M)\}_{k=1}^n$ the eigenvalues of a hermitian matrix in increasing order.
With this we have:

$$|\lambda_k(A) - \lambda_k(B)| \leqslant ||A - B||.$$

Proof. After the redefinition $B \mapsto (-B)$, what we need to prove is

$$|\lambda_i(A + B) - \lambda_i(A)| \leqslant ||B||$$

for Hermitian A, B. Since we have

$$\lambda_i(A) - \lambda_i(A + B) = \lambda_i((A + B) + (-B)) - \lambda_i(A + B)$$

and $|| - B|| = ||B||$ it follows that it suffices to prove

$$\lambda_i(A + B) - \lambda_i(A) \leqslant ||B||$$

for arbitrary hermitian A, B.

We note that the Courant-Fischer $\min - \max$ theorem tells us that if A is an $n \times n$ Hermitian matrix, we have

$$\lambda_i(M) = \sup_{\dim(V)=i} \inf_{v \in V, ||v||=1} v^* M v.$$

With this we find

$$\begin{aligned}
\lambda_i(A+B)-\lambda_i(A) &= \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*(A+B)v - \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Av\\
&\leqslant \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Av + \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Bv\\
&- \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Av\\
&= \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Bv\\
&= \sup_{\dim(V)=i}\inf_{v\in V,||v||=1} v^*Bv\\
&\leqslant \max_{1\leqslant k\leqslant n}\{|\lambda_k(B)|\}\\
&= ||B||.
\end{aligned}$$

□

K.1.3 Symmetric Normalizations

Most common spectral graph convolutional networks (such as e.g. He et al. (2021); Bianchi et al. (2019); Defferrard et al. (2016)) base the learnable filters that they propose on the symmetrically normalized graph Laplacian

$$\mathscr{L} = Id - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}.$$

In the setting $S_{\text{high}} \gg S_{\text{reg.}}$ we are considering, this leads to an effective feature propagation along edges in $\mathcal{E}_{\text{high}}$ and $\mathcal{E}_{\text{low, exclusive}}$ (c.f. also Fig. 21) only, as Theorem K.3 below establishes:

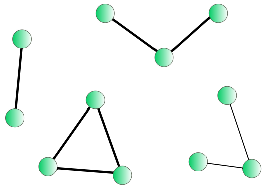

Figure 21: Limit graph corresponding to Fig 19 for symmetric normalization

Theorem K.3. With

$$T = Id - D^{-\frac{1}{2}}WD^{-\frac{1}{2}},$$

and the scale decomposition as introduced above, we have that

$$\left\|T - \left(Id - D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} - D_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}}\right)\right\| = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right) \tag{21}$$

for $S_{\text{high}} \gg S_{\text{reg.}}$.

Proof. We first note that instead of (21), we may equivalently establish

$$\left\|D^{-\frac{1}{2}}WD^{-\frac{1}{2}} - \left(D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}}\right)\right\| = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right).$$

We have

$$W = W_{\text{high}} + W_{\text{reg.}}.$$

With this, we may write

$$D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}} + D^{-\frac{1}{2}}W_{\text{reg.}}D^{-\frac{1}{2}}. \tag{22}$$

Let us first examine the term $D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}}$. We note for the corresponding matrix entries that

$$\left(D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{d_i}}\cdot(W_{\text{high}})_{ij}\cdot\frac{1}{\sqrt{d_j}}$$

Let us use the notation

$$d_i^{\text{high}} = \sum_{j=1}^{N}(W_{\text{high}})_{ij}, \quad d_i^{\text{reg.}} = \sum_{j=1}^{N}(W_{\text{reg.}})_{ij} \text{ and } d_i^{\text{low,exclusive}} = \sum_{j=1}^{N}(W_{\text{low,exclusive}})_{ij}.$$

We then find

$$\frac{1}{\sqrt{d_i}} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}}$$

Using the Taylor expansion

$$\frac{1}{\sqrt{1+\epsilon}} = 1 - \frac{1}{2}\epsilon + \mathcal{O}(\epsilon^2),$$

we thus have

$$\left(D^{-\frac{1}{2}} W_{\text{high}} D^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} + \mathcal{O}\left(\frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}\right).$$

Since we have

$$\frac{d_i^{\text{reg.}}}{d_i^{\text{high}}} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}},$$

this yields

$$D^{-\frac{1}{2}} W_{\text{high}} D^{-\frac{1}{2}} = D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right).$$

Thus let us turn towards the second summand on the right-hand-side of (22). We have

$$\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{d_i}} \cdot (W_{\text{reg.}})_{ij} \cdot \frac{1}{\sqrt{d_j}}.$$

Suppose that either i or j is not in $G_{\text{low, exclusive}}$. Without loss of generality (since the matrix under consideration is symmetric), assume $i \notin G_{\text{low, exclusive}}$, but $(W_{\text{reg.}})_{ij} \neq 0$. We may again write

$$\frac{1}{\sqrt{d_j}} = \frac{1}{\sqrt{d_j^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}}.$$

Since

$$\frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}} \leqslant 1,$$

we have

$$\left|\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij}\right| \leqslant \left|\frac{1}{\sqrt{d_i}} \cdot (W_{\text{reg.}})_{ij}\right| \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right).$$

If instead we have $i, j \in G_{\text{low, exclusive}}$, then clearly

$$\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij} = \left(D_{\text{reg.}}^{-\frac{1}{2}} W_{\text{low,exclusive}} D_{\text{reg.}}^{-\frac{1}{2}}\right)_{ij}.$$

Thus in total we have established

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = \left(D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}} W_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}}\right) + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right)$$

which was to be established.

□

Apart from networks that make use of the symmetrically normalized graph Laplacian $\mathscr{L}$, some methods, such as most notably Kipf & Welling (2017), instead base their filters on the operator

$$T = \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}},$$

with

$$\tilde{W} = (W + Id)$$

and

$$\tilde{D} = D + Id.$$

In analogy to Theorem K.3, we here establish the limit propagation scheme determined by such operators:

Theorem K.4. With
$$T = \tilde{D}^{-\frac{1}{2}}\tilde{W}\tilde{D}^{-\frac{1}{2}},$$
where $\tilde{W} = (W + Id)$ and $\tilde{D} = D + Id$ as well as the scale decomposition introduced above, we have that

$$\left\| T - \left(D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}} \tilde{W}_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}} \right) \right\| = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}} + 1}{S_{\text{high}}}} \right)$$

for $S_{\text{high}} \gg S_{\text{reg.}}$. Here $\tilde{W}_{\text{low, exclusive}}$ is given as

$$\tilde{W}_{\text{low, exclusive}} := W_{\text{low, exclusive}} + \text{diag}\left(\mathbb{1}_{G_{\text{low, exclusive}}} \right)$$

and $\mathbb{1}_{G_{\text{low, exclusive}}}$ denotes the vector whose entries are one for nodes in $G_{\text{low, exclusive}}$ and zero for all other nodes.

The difference to the result of Theorem K.3 is thus that applicability of the limit propagation scheme of Fig. 21 for the GCN Kipf & Welling (2017) is not only contingent upon $S_{\text{high}} \gg S_{\text{reg.}}$ but also $S_{\text{high}} \gg 1$.

Proof. To establish this – as in the proof of Theorem K.3 – we first decompose T:

$$\begin{aligned} \tilde{D}^{-\frac{1}{2}}\tilde{W}\tilde{D}^{-\frac{1}{2}} &= \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} Id \tilde{D}^{-\frac{1}{2}} \\ &= \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-1} \end{aligned} \tag{23}$$

For the first term, we note

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i + 1}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j + 1}}.$$

We then find
$$\frac{1}{\sqrt{d_i + 1}} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}} + 1}{d_i^{\text{high}}}}}.$$

Analogously to the proof of Theorem K.3, this yields

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} + \mathcal{O}\left(\frac{1 + d_i^{\text{reg.}}}{d_i^{\text{high}}} \right).$$

This implies
$$\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} = D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + \mathcal{O}\left(\frac{S_{\text{reg.}} + 1}{S_{\text{high}}} \right).$$

Next we turn to the second summand in (23):

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i + 1}} \cdot (W_{\text{reg.}})_{ij} \cdot \frac{1}{\sqrt{d_j + 1}}.$$

Suppose that either i or j is not in $G_{\text{low, exclusive}}$. Without loss of generality (since the matrix under consideration is symmetric), assume $i \notin G_{\text{low, exclusive}}$, but $(W_{\text{reg.}})_{ij} \neq 0$. We may again write

$$\frac{1}{\sqrt{d_j + 1}} = \frac{1}{\sqrt{d_j^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}} + 1}{d_i^{\text{high}}}}}.$$

Since
$$\frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}} + 1}{d_i^{\text{high}}}}} \leqslant 1,$$

we have

$$\begin{aligned}\left|\left(D^{-\frac{1}{2}}W_{\text{reg.}}D^{-\frac{1}{2}}\right)_{ij}\right| &\leqslant \left|\frac{1}{\sqrt{1+d_i}}\cdot (W_{\text{reg.}})_{ij}\right|\cdot\frac{1}{\sqrt{d_j^{\text{high}}}}\\ &\leqslant \left|\frac{1}{\sqrt{d_i^{\text{reg.}}}}\cdot (W_{\text{reg.}})_{ij}\right|\cdot\frac{1}{\sqrt{d_j^{\text{high}}}}\\ &= \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right).\end{aligned}$$

If instead we have $i, j \in G_{\text{low, exclusive}}$, then clearly

$$\left(\tilde{D}^{-\frac{1}{2}}W_{\text{reg.}}\tilde{D}^{-\frac{1}{2}}\right)_{ij} = \left(\tilde{D}_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low,exclusive}}\tilde{D}_{\text{reg.}}^{-\frac{1}{2}}\right)_{ij}.$$

Finally we note for the third term on the right-hand-side of (23) that

$$\frac{1}{d_i} \leqslant \frac{1}{d_i^{\text{high}}} = \mathcal{O}\left(\frac{1}{S_{\text{high}}}\right)$$

if $i \notin G_{\text{low, exclusive}}$.

In total we thus have found

$$\tilde{D}^{-\frac{1}{2}}\tilde{W}\tilde{D}^{-\frac{1}{2}} = \left(D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}}\tilde{W}_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}}\right) + \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}+1}{S_{\text{high}}}}\right);$$

which was to be proved. □

K.2 Spatial Convolutional Filters

Apart from spectral methods, there of course also exist methods that purely operate in the spatial domain of the graph. Such methods most often fall into the paradigm of message passing neural networks (MPNNs) Gilmer et al. (2017); Fey & Lenssen (2019): With $X_i^\ell \in \mathbb{R}^F$ denoting the features of node i in layer ℓ and w_{ij} denoting edge features, a message passing neural network may be described by the update rule (c.f. Gilmer et al. (2017))

$$X_i^{\ell+1} = \gamma\left(X_i^\ell, \coprod_{j\in\mathcal{N}(i)}\phi\left(X_i^\ell, X_j^\ell, w_{ij}\right)\right). \tag{24}$$

Here $\mathcal{N}(i)$ denotes the neighbourhood of node i, $\coprod$ denotes a differentiable and permutation invariant function (typically "sum", "mean" or "max") while γ and ϕ denote differentiable functions such as multi-layer-perceptrons (MLPs) which might not be the same in each layer. Fey & Lenssen (2019).

Before we discuss corresponding limit-propagation schemes, we first establish that MPNNs are not able to reproduce the limit propagation scheme ofFigure 6 (b) and are thus not stable to scale transitions and topological perturbations.

K.2.1 Scale-Sensitivity of Message Passing Neural Networks

Here we establish that message passing networks (as defined in (24) above) are unable to emulate a limit propagation scheme similar to the one in Figure 6 (b). Hence such architectures are also not stable to scale-changing topological perturbations such as coarse-graining procedures.

To this end, we consider a simple, fully connected graph G on three nodes labeled 1, 2 and 3 (c.f. Fig. 22). We assume all node-weights to be equal to one ($\mu_i = 1$ for $i = 1, 2, 3$) and edge weights

$$w_{13}, w_{23} \leqslant S_{\text{reg.}}$$

as well as

$$w_{12} = S_{\text{high}}.$$

We now assume $S_{\text{high}} \gg S_{\text{reg.}}$.

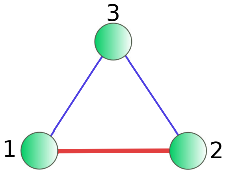

Figure 22: Three node Graph G with on large weight $w_{12} \gg 1$.

Given states $\{X_1^\ell, X_2^\ell, X_3^\ell\}$ in layer ℓ, a limit propagation scheme as in Figure 6 (b) would require the updated feature vector of node 3 to be given by

$$X_{3,\text{desired}}^{\ell+1} := \gamma\left(X_3^\ell, \phi\left(X_3^\ell, \frac{X_1^\ell + X_2^\ell}{2}, (w_{31} + w_{32})\right)\right)$$

However, the actual updated feature at node 3 is given as (c.f. (24)):

$$X_{3,\text{actual}}^{\ell+1} := \gamma\left(X_3^\ell, \phi\left(X_3^\ell, X_1^\ell, w_{31}\right) \coprod \phi\left(X_3^\ell, X_2^\ell, w_{32}\right)\right) \tag{25}$$

Since there is no dependence on S_{high} in equation (25) – which defines $X_{3,\text{actual}}^{\ell+1}$ – the desired propagation scheme can not arise, unless it is paradoxically already present at all scales S_{high}. If it is present at all scales, there is however only propagation along edges in $\underline{G}$, even if $S_{\text{high}} \approx S_{\text{reg.}}$, which would imply that the message passing network would not respect the graph structure of G. Hence $X_{3,\text{actual}}^{\ell+1} \nrightarrow X_{3,\text{desired}}^{\ell+1}$ does not converge as S_{high} increases.

K.2.2 Limit Propagation Schemes

The number of possible choices of message functions ϕ, aggregation functions $\coprod$ and update functions γ is clearly endless. Here we shall exemplarily discuss limit propagation schemes for two popular architectures: We first discuss the most general case where the message function ϕ is given as a learnable perceptron. Subsequently we assume that node features are updated with an attention-type mechanism.

Generic message functions: We first consider the possibility that the message function ϕ in (25) is implemented via an MLP using ReLU-activations: Assuming (for simplicity in notation) a one-hidden-layer MLP mapping features $X_i^\ell \in \mathbb{R}^{F_\ell}$ to features $X_i^{\ell+1} \in \mathbb{R}^{F_{\ell+1}}$ we have

$$\phi(X_i^\ell, X_j^\ell, w_{ij}) = \text{ReLU}\left(W_1^\ell \cdot X_i^\ell + W_2^\ell \cdot X_2^\ell + W_3^\ell \cdot w_{ij} + B^\ell\right)$$

with bias term $B^{\ell+1} \in \mathbb{R}^{F_{\ell+1}}$ and weight matrices $W_1^{\ell+1}, W_2^{\ell+1} \in \mathbb{R}^{F_{\ell+1} \times F_\ell}$ and $W_3^\ell \in \mathbb{R}^{F_{\ell+1}}$.

We will assume that the weight-vecor $W_3^{\ell+1}$ has no-nonzero entries. This is not a severe limitation experimentally and in fact generically justified: The complementary event of at-least one entry of W_3 being assigned precisely zero during training has probability weight zero (assuming an absolutely continuous probability distribtuion according to which weights are learned).

Let us now assume that the edge (ij) belongs to $\mathcal{E}_{\text{high}}$ and the corresponding weight w_{ij} is large ($w_{ij} \gg 1$). The behaviour of entries $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ of the message $\phi(X_i^\ell, X_j^\ell, w_{ij}) \in \mathbb{R}^{F_{\ell+1}}$ is then determined by the sign of the corresponding entry $\left(W_3^\ell\right)_a$ of the weight vector $W_3^\ell \in \mathbb{R}^{F_{\ell+1}}$:

If we have $\left(W_3^\ell\right)_a < 0$, then $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ approaches zero for larger edge-weights w_{ij}:

$$\lim_{w_{ij} \to \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = 0 \tag{26}$$

If we have $\left(W_3^\ell\right)_a > 0$, then $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ increasingly diverges for larger edge-weights w_{ij}:

$$\lim_{w_{ij} \to \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \infty \tag{27}$$

For either choice of aggregation function $\coprod$ in (24) among "max", "sum" or "mean" the behaviour in (27) leads to unstable networks if the update function γ is also given as an MLP with ReLU activations. Apart from instabilities, we also make the following observation: If $S_{\text{high}} \gg S_{\text{reg.}}$, then by (27) and continuity of ϕ we can conclude that components $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ of messages propagated along $\mathcal{E}_{\text{high}}$ for which $\left(W_3^\ell\right)_a > 0$ dominate over messages propagated along edges in $\mathcal{E}_{\text{reg.}}$. By (26), the former clearly also dominate over components $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ of messages propagated along $\mathcal{E}_{\text{high}}$ for which $\left(W_3^\ell\right)_a < 0$. This behaviour is irrespective of whether "max", "sum" or "mean" aggregations are employed. Hence the limit propagation scheme essentially only takes into account message channels $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ for which $(ij) \in \mathcal{E}_{\text{high}}$ and $\left(W_3^\ell\right)_a > 0$.

Similar considerations apply, if non-linearities are chosen as leaky ReLU. If instead of ReLU activations a sigmoid-nonlinearity σ like $\tanh$ is employed, messages propagated along $\mathcal{E}_{\text{large}}$ become increasingly uninformative, since they are progressively more independent of features X_i^ℓ and weights w_{ij}. Indeed, for sigmoid activations, the limits (26) and (27) are given as follows:

If we have $\left(W_3^\ell\right)_a < 0$, then we have for larger edge-weights w_{ij} that

$$\lim_{w_{ij}\to\infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \lim_{y\to-\infty} \sigma(y).$$

If we have $\left(W_3^\ell\right)_a > 0$, then

$$\lim_{w_{ij}\to\infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \lim_{y\to\infty} \sigma(y).$$

In both cases, the messages $\phi(X_i^\ell, X_j^\ell, w_{ij})$ propagated along $\mathcal{E}_{\text{large}}$ become increasingly constant as the scale S_{high} increases.

Attention based messages: Apart from general learnable message functions as above, we here also discuss an approach where edge weights are re-learned in an attention based manner. For this we modify the method Velickovic et al. (2018) to include edge weights. The resulting propagation scheme – with a single attention head for simplicity and a non-linearity ρ – is given as

$$X_i^{\ell+1} = \rho\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}(WX_j^{\ell+1})\right).$$

Here we have $W \in \mathbb{R}^{F_{\ell+1}\times F_\ell}$ and

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_j^\ell \,\|\, w_{ij}\right]\right)\right)}{\sum\limits_{k\in\mathcal{N}(i)} \exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_k^\ell \,\|\, w_{ik}\right]\right)\right)}, \tag{28}$$

with $\|$ denoting concatenation. The weight vector $\vec{a} \in \mathbb{R}^{2F_{\ell+1}+1}$ is assumed to have a non zero entry in its last component. Otherwise, this attention mechanism would correspond to the one proposed in Velickovic et al. (2018), which does not take into account edge weights. Let us denote this entry of $\vec{a}$ ()determining attention on the weight w_{ij}) by a_w.

If $a_w < 0$, we have for $(i,j) \in \mathcal{E}_{\text{high}}$ that

$$\exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_j^\ell \,\|\, w_{ij}\right]\right)\right) \longrightarrow 0$$

as the weight w_{ij} increases. Thus propagation along edges in $\mathcal{E}_{\text{high}}$ is essentially suppressed in this case.

If $a_w > 0$, we have for $(i,j) \in \mathcal{E}_{\text{high}}$ that

$$\exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_j^\ell \,\|\, w_{ij}\right]\right)\right) \longrightarrow \infty$$

as the weight w_{ij} increases. Thus for edges $(i,j) \in \mathcal{E}_{\text{reg.}}$ (i.e. those that are *not* in $\mathcal{E}_{\text{high}}$), we have

$$\alpha_{ij} \to 0,$$

since the denominator in (28) diverges. Hence in this case, propagation along $\mathcal{E}_{\text{reg.}}$ is essentially suppressed and features are effectively only propagated along $\mathcal{E}_{\text{high}}$.