
Graph Neural Networks Are Not Continuous Across Graph Resolutions

Christian Koke^{1 2 3} Yuesong Shen^{2 3} Abhishek Saroha^{2 3} Marvin Eisenberger³ Bastian Rieck^{4 5}
Michael Bronstein^{1 6} Daniel Cremers^{2 3}

Abstract

We show that contrary to conventional wisdom in the community, graph neural networks (GNNs) are not continuous with respect to all natural modes of graph convergence. As a result, GNNs may generate substantially different latent representations for graphs that are very similar. In particular they assign vastly different latent embeddings to graphs that represent the same underlying object at different resolution scales. We trace this failure of continuity back to a structural obstruction arising from commonly used information-propagation schemes. Building on this insight, we then derive a principled modification to standard GNN architectures which equips models with continuity across scales. The proposed modification enables consistent integration of distinct resolutions and reliable generalization between them. We systematically validate our theoretical findings in a wide range of numerical experiments. Our code is available at <https://github.com/christiankoke/scale-continuous-gnns>.

1. Introduction

Graph Neural Networks (GNNs) (Scarselli et al., 2009; Bruna et al., 2014; Gilmer et al., 2017) have become the dominant framework for learning on graph structured data and have achieved widespread success in domains as varied as protein structure prediction (Jumper et al., 2021), material science (Xie & Grossman, 2018), weather forecasting (Lam et al., 2023) or catalyst screening (Price et al., 2022).

A key theoretical pillar supporting their widespread adoption is the commonly held belief that GNNs define *stable*

and *continuous* mappings: small changes in the input graph are expected to induce correspondingly small changes in learned representations. Indeed, a substantial body of prior work has established continuity, stability, and robustness for GNNs under a broad range of perturbation modes: Previous investigations e.g. considered GNN-stability under small variations in edge-weights (Levie et al., 2019b; Gama et al., 2020) or graph rewiring (Kenlay et al., 2021). Consistency of latent representations has also been established across graphs that faithfully approximate the same metric measure space (Levie et al., 2019a), graphon (Ruiz et al., 2020; Maskey et al., 2021), graphing (Roddenberry et al., 2022) or graphop (Le & Jegelka, 2023).¹ Taken together, these results have contributed to a prevailing view in the field that graph neural networks define broadly stable and well-behaved mappings: *similar input graphs are mapped to similar latent embeddings*.

In this work, we show that this is, however, not always the case: In particular we demonstrate that standard GNN architectures can produce *drastically different latent representations* for graphs that describe the same underlying object at different *resolution scales*. As a consequence, models trained on graphs at one resolution, for instance, *fail* to generalize to equivalent graphs at another resolution.

We trace this failure to the previously understudied fact that graph neural networks are generally *not* continuous with respect to all natural modes of graph convergence: We observe that there is a natural interpolating sequence of graphs starting at an original graph and terminating at its coarse-grained version. Mathematically, this sequence smoothly interpolates between the original- and coarse heat kernel. This convergence is not respected by GNNs: Even though input graphs converge (in the heat sense) the generated sequence of latent embeddings does not converge.

This observation reveals a fundamental limitation of existing GNN architectures. It also motivates the development of modifications to existing architectures, in order to render them *continuous* across graph resolution scales.

Concretely, our paper makes the following **contributions**:

¹Appendix B further summarizes these existing perturbation models, also discussing their inapplicability to our setting.

¹AITHYRA ²Munich Center for Machine Learning ³Technical University of Munich, Munich ⁴University of Fribourg ⁵Institute of Computational Biology ⁶University of Oxford. Correspondence to: Christian Koke <christian.koke@tum.de>.

- We identify and empirically demonstrate a previously underappreciated failure mode of standard GNNs, i.e., **a lack of continuity across graph resolutions**.
- We provide a **theoretical explanation** linking this failure mode to discontinuity under graph convergence.
- We show that the **underlying obstruction is structural**, originating in standard propagation-schemes.
- We show how to **modify propagation schemes** to yield **provably scale-continuous GNN architectures**.
- We establish **theoretical guarantees** and present **extensive empirical validation** demonstrating our modifications facilitate reliable **generalization across scales**.

2. Graphs and Graph Learning

We first briefly review graphs and graph neural networks:

2.1. Graphs and their Description

We consider finite, weighted graphs $G = (V, E)$ equipped with a *node weight matrix* $M = \text{diag}(\mu_v)_{v \in V}$. Each weight $\mu_v > 0$ reflects node-wise information. Examples are physical mass or charge of atom v in a molecular graph, number of users represented if v is a node in a social network, or a volume element if v is part of a point cloud. Often, all node masses are set to $\mu_v = 1$ so that $M = I$. Edges $(u, v) \in E$ of a graph carry symmetric non-negative (potentially binary) weights A_{uv} which we collect in the adjacency matrix A . The corresponding degree matrix is $D = \text{diag}(d_v)$ with $d_v = \sum_u A_{uv}$.

The natural self-adjoint Laplacian associated with a node-weighted graph is $L = M^{-1}(D - A)$. It is self-adjoint with respect to the M -weighted inner product $\langle f, g \rangle_M = f^\top M g$. When the node weights encode local volumes or sampling densities it may be thought of as the discrete counterpart of the Laplacian $L = -\Delta$ on a continuum domain (Chung, 1997). While the natural Laplacian is central to our work, more commonly used in machine learning applications is instead the normalized Laplacian $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$. It is self-adjoint in the standard Euclidean inner product and its spectrum is contained in the interval $[0, 2]$.

2.2. Graph Neural Networks

Graph neural networks are machine learning methods adapted to handle graph structured data. Corresponding network architectures broadly fall into two categories:

Message Passing Networks: Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017; Bronstein et al., 2021) are arguably the most widely used class of GNNs. At

each layer k , node features $h_v^{(k)}$ are updated by aggregating information from neighbouring nodes:

$$m_{vu}^{(k)} = \phi(h_v^{(k)}, h_u^{(k)}, A_{uv}) \quad (1)$$

$$h_v^{(k+1)} = \text{AGG}_{u \in \mathcal{N}(v)} m_{vu}^{(k)}. \quad (2)$$

Here A_{uv} denotes edge weights. Architectures differ in which message- and aggregation functions are being used (Xu et al., 2019; Velickovic et al., 2018; Kipf & Welling, 2017; Hamilton et al., 2017), with typical choices for the neighborhood aggregation function AGG being sum, avg, or (seldom) max (Fey et al., 2025).

Spectral Networks: Spectral graph convolutional layers (Bruna et al., 2014; Defferrard et al., 2016) apply learnable filters $g_\theta(\mathcal{L})$ to node-wise information via matrix-multiplication. The filters themselves are composed of learnable scalar functions $g_\theta(\cdot)$ which are applied to the normalized graph Laplacian $\mathcal{L}$. In practice, the filter g_θ is not implemented via eigendecomposition but approximated via polynomials (He et al., 2022; 2021; Koke & Cremers, 2024; Levie et al., 2019c). The prototypical example is ChebNet, where filters are parametrized as $g_\theta(\lambda) \approx \sum_{\ell=0}^K \theta_\ell T_\ell(\lambda)$, with T_ℓ the ℓ^{th} -Chebyshev polynomial. With node feature matrix X and learnable weight matrices $\{W_\ell\}_\ell$, a spectral layer is then efficiently implemented as

$$X^{(k+1)} = \sum_{\ell=0}^K T_\ell(\mathcal{L}) X^{(k)} W_\ell. \quad (3)$$

3. The Failure Mode: Lack of Generalization

To showcase how standard graph neural networks are unable to consistently integrate varying scales experimentally, we make use of the QM7 dataset (Rupp et al., 2012), consisting of organic molecules containing both hydrogen and heavy atoms. Prediction target is the 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 energies between atoms i, j , with $|\vec{x}_i - \vec{x}_j|$ denoting the interatomic distance and Z_i atomic charge. 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 various atomic nuclei are discarded.² Instead, only an aggregate description is used wherein each nucleus is only described by a single node.

To test the ability of GNNs to generalize across scales, we additionally also consider a version of QM7 where we lower the resolution scale even further: To this end we aggregate each heavy atomic core together with its surrounding (single-proton) hydrogen atoms into super-nodes. As depicted in

²At an even higher resolution scale, also interactions of quarks and gluons (in turn making up protons and neutrons) are neglected.

Fig. 1, this is done via standard graph coarsification (Loukas & Vandenbrouck, 2018b; Loukas, 2019); Appendix G.1 provides exact details.

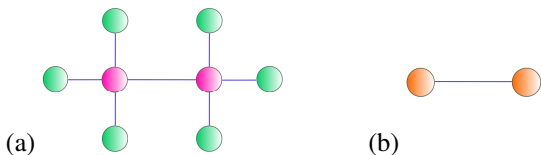


Figure 1. (a) Original graph G corresponding to the Ethane molecule with Carbon in purple and Hydrogen in green (b) Coarse grained $\underline{G}$ with aggregate Carbon-Hydrogen super-nodes in orange

This $\text{QM7}_{\text{coarse}}$ dataset models data obtained from a resolution-limited observation process unable to resolve positions of individual (small) hydrogen atoms and only providing information about how many are bound to a given heavy atom. Using the high-resolution graphs $\{G\}$ of QM7 and the low-resolution graphs $\{\underline{G}\}$ in $\text{QM7}_{\text{coarse}}$, we then investigate the ability of GNNs³ to consistently handle various scales 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) during inference increase significantly, when going from a same-resolution setting to a cross-resolution setting. *None of the standard architecture types are able to consistently handle more than one scale.* Also employing multi-scale propagation schemes (SAG, SAG-M, UFGNet, Lanczos) does not result in consistently incorporating multiple scales: Corresponding cross-resolution MAEs are among the largest.

Table 1. QM7 regression. Mean Absolute Error (MAE $\downarrow$) in kcal/mol for training and inference at different resolutions scales.

Resolution	MAE ($\downarrow$) on QM7 [kcal/mol]			
	High		Low	
Training:	High		Low	
Inference:	Low	High	Low	High
GCN	136.7 ± 6.6	(63.6 ± 1.3)	(63.6 ± 1.3)	138.1 ± 2.4
GATv2	423.5 ± 337.1	(67.4 ± 8.2)	(59.7 ± 2.7)	257.4 ± 139.1
ChebNet	447.8 ± 6.0	(66.7 ± 1.4)	(71.5 ± 2.1)	158.7 ± 57.4
GIN	658.4 ± 85.8	(17.2 ± 2.8)	(38.3 ± 21.6)	1835.4 ± 925.8
SAG	589.7 ± 44.9	(68.2 ± 2.6)	(107.9 ± 1.1)	283.6 ± 39.3
SAG-M	194.4 ± 29.9	(66.6 ± 1.9)	(77.8 ± 6.0)	219.5 ± 11.7
UFGNet	131.5 ± 6.1	(62.4 ± 0.7)	(69.4 ± 0.7)	148.1 ± 6.3
Lanczos	938.4 ± 2.5	(9.9 ± 2.5)	(88.2 ± 2.7)	658.6 ± 199.2

We can trace the inability of common models to generalize back to the difference in latent embeddings $\{F\}$ and $\{\underline{F}\}$, generated for original $\{G\}$ and coarsified graphs $\{\underline{G}\}$.

As evident from Table 2, latent embeddings of graphs describing the same object on varying resolutions are significantly different. This in turn explains the inability to

³Appendix G.1 contains further details on models.

Table 2. Embedding difference $\|F - \underline{F}\|$ across resolution scales averaged over 5 runs (mean $\pm$ std). Lower is better ($\downarrow$).

Training $\rightarrow$ Inference:	High $\rightarrow$ Low	Low $\rightarrow$ High
GCN	20.3 $\pm$ 1.6	18.0 $\pm$ 0.6
GATv2	76.8 $\pm$ 52.4	42.5 $\pm$ 24.3
ChebNet	123.0 $\pm$ 3.8	158.5 $\pm$ 57.40
GIN	881.2 $\pm$ 268.7	2875.8 $\pm$ 1527.7
SAG	113.5 $\pm$ 10.3	43.3 $\pm$ 5.4
SAG-M	75.0 $\pm$ 15.1	61.7 $\pm$ 4.6
UFGNet	24.5 $\pm$ 0.3	26.8 $\pm$ 2.1
Lanczos	1286.1 $\pm$ 54.4	728.4 $\pm$ 230.0

generalize across scales. It should be noted that in practice, this failure *cannot* be overcome by augmenting the training set, as we have no way of generating faithful high-resolution descriptions given only lower resolution training data.

4. The Underlying Problem: Discontinuity

Our next goal then is to understand *why* the latent embeddings $F, \underline{F}$ generated for original (G) and coarse ($\underline{G}$) graph descriptions of the same underlying object are so vastly different. Counter to conventional wisdom and consensus within the graph learning community we find below that

GNNs are not continuous across scales.

Hence, they may produce vastly different outputs ($F, \underline{F}$) for highly similar inputs ($G, \underline{G}$).

To see this discontinuity in action, note that in the coarse graphs $\{\underline{G}\}$ of $\text{QM7}_{\text{coarse}}$, hydrogen atoms have been fused into their nearest heavy atoms. Such a coarse graph hence can be viewed as the limit of a process in which hydrogen atoms are displaced from equilibrium toward their nearest heavy atom. The limit graph is obtained once they are fully merged into the corresponding heavy nodes.

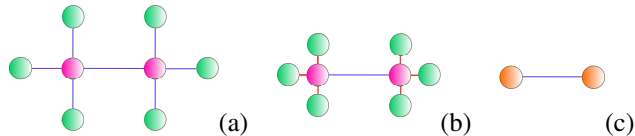


Figure 2. Collapsing Procedure visualized

If standard GNN architectures were continuous, the convergence of this graph modification process in Fig. 2 towards a limit graph should be reflected also in the latent space: Latent embeddings of modified graphs should converge to the latent embedding of the limit graph.

Figure 3 compares embeddings $\{\underline{F}\}$ generated for coarsified graphs $\{\underline{G}\}$, with embeddings $\{F_\omega\}$ of graphs $\{G_\omega\}$ where hydrogen atoms have been moved to reduce the distance towards their nearest heavy atoms by a factor of $\omega \geq 1$ (i.e.

$\text{dist}_{\text{new}} = \text{dist}_{\text{equilib.}}/\omega$), but have not yet completely arrived at the positions of nearest heavy atoms.

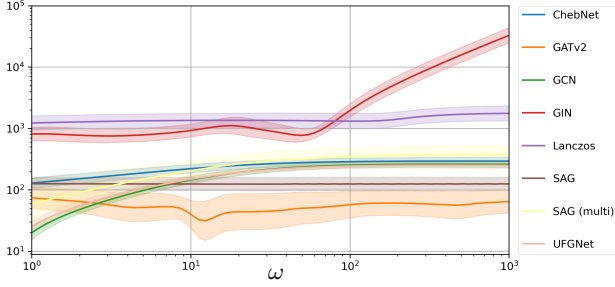


Figure 3. Latent distance $\|F_\omega - \underline{F}\|$

From Fig. 3, we observe that latent embeddings do *not* converge ($\|F_\omega - \underline{F}\| \not\rightarrow 0$). Since the sequence $\{G_\omega\}_\omega$ of input graphs converges⁴ but GNN outputs evidently do not, they indeed cannot be considered continuous. Hence they may map similar graphs to vastly different latent embeddings.

5. The Obstruction: GNN Propagation

We can understand the underlying reason for this discontinuity by exemplarily investigating the prototypical graph neural network GCN (Kipf & Welling, 2017):⁵ Inside a GCN-layer, a node feature matrix $X \in \mathbb{R}^{N \times F}$ (with number of nodes N and feature dimension F) is updated as

$$X \mapsto \hat{A}XW. \quad (4)$$

Here $W \in \mathbb{R}^{F \times F}$ mixes channels. Information flow over the graph is implemented via the so called *renormalized* adjacency matrix $\hat{A} \in \mathbb{R}^{N \times N}$, with $\hat{A}_{ij} \sim A_{ij}/\sqrt{d_i d_j}$. When moving hydrogen (H) atoms towards heavy atoms ($|\vec{x}_H - \vec{x}_{\text{heavy}}| \rightarrow 0$), corresponding edge weights $A_{H,\text{heavy}} = 1 \cdot Z_{\text{heavy}} \cdot |\vec{x}_H - \vec{x}_{\text{heavy}}|^{-1}$ of the *original* adjacency matrix A tend to infinity. Thus also node-degrees associated to heavy atoms tend to infinity. Since distances (and hence weights in the adjacency matrix A) between heavy atoms remain constant, the “renormalized” entries $\hat{A}_{\text{heavy},\text{heavy}}$ in $\hat{A}$, however, tend to zero instead.

Hence, as hydrogen atoms are moved out of equilibrium towards their final positions, the communication between heavy atoms in the modified graphs G_ω becomes severely disrupted ($\hat{A}_{\text{heavy},\text{heavy}} \rightarrow 0$). Information is only propagated along a severely disconnected effective limit graph (dissected into distinct connected components (Fig 4 (a)); and not along the true coarse limit graph $\underline{G}$ (Fig. 4 (b)).

⁴Section 6.1 discusses the precise sense in which this convergence is formalized mathematically.

⁵Appendix D contains derivations for all standard GNN types.

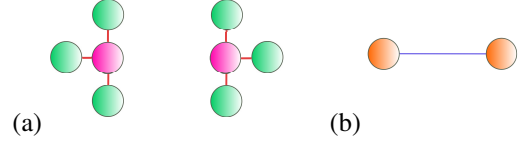


Figure 4. (a) Effective propagation vs. (b) true coarse graph $\underline{G}$

As a consequence of the information flows over the graphs $G_\omega, \underline{G}$ being vastly different, the latent embeddings $F_\omega, \underline{F}$ that are being generated for the two graphs differ greatly.

At first glance, it may seem that the observations above apply only to the QM7 dataset. There edge weights scale inversely with distance, so sequences of graphs with diverging weights arise naturally. In Section 6 immediately below, however, we discuss that this is merely a proxy for a general fact in graph theory: the natural interpolation between an original graph G and its coarsified version $\underline{G}$ proceeds by letting the edge weights within those clusters of G that collapse to the super-nodes of $\underline{G}$ tend to infinity (Koke, 2026a;b). Thus the observations above are indeed applicable to *generic* datasets.

6. Solution: Laplace Transform Propagation

Having established that standard GNNs are not continuous across scales due to the way they propagate information over a given graph G , let us now work towards building architectures that instead *are* continuous.

6.1. From Graphs to Heat Kernels

To this end, let us begin by formalizing rigorously, in which sense the sequence of graphs G_ω considered in Section 5 approaches the coarsified limit graph $\underline{G}$:

Heat Kernel convergence: When moving hydrogen atoms out of equilibrium, we are significantly increasing certain weights ($A_{H,\text{heavy}} \sim |\vec{x}_H - \vec{x}_{\text{heavy}}|^{-1} \sim \omega \rightarrow \infty$). From a heat diffusion perspective, information in a graph equalizes much faster along edges with very large weights. In the limit where edge-weights within certain sub-graphs tend to infinity, information within these clusters equalizes immediately. Each such sub-graph thus effectively behaves as a single node in a coarse grained effective graph $\underline{G}$.

To quantify this, we recall that the diffusion equation on a graph is given by $dX(t)/dt = -L \cdot X(t)$ with solution $X(t) = e^{-Lt} \cdot X(0)$. As was established rigorously in Koke (2026b) (cf. Appendix H for a discussion) the convergence

$$\|e^{-tL_\omega} - J^\dagger e^{-t\underline{L}} J^\downarrow\| \rightarrow 0 \quad (5)$$

then holds for any fixed $t > 0$ as $\omega \rightarrow \infty$. Here $L_\omega, \underline{L}$ are the Laplacians of the respective graphs $G_\omega, \underline{G}$. The matrices $J^{\downarrow,\uparrow}$ appear naturally and allow to compare graphs of different sizes: Projection $J^\downarrow$ assigns the average over

strongly connected clusters to the super-node representing this cluster in $\underline{G}$. Interpolation $J^\uparrow$ assigns information at a super-node in $\underline{G}$ to each node in the respective cluster in G .

It is important to note that in order to detect the convergence $G_\omega \rightarrow \underline{G}$ as in (5) one needs to make use of the (non-normalized) Laplacian L . Using e.g. the normalized Laplacian $\mathcal{L}$ would not lead to convergence ($e^{-t\mathcal{L}_\omega} \not\rightarrow J^\uparrow e^{-t\mathcal{L}} J^\downarrow$). Also directly comparing the Laplacians L, L_ω would not detect the convergence: Clearly $\|L_\omega\| \rightarrow \infty$, so that the sequence of Laplacians L_ω alone does not converge.

Interpolating between Graphs on different scales: We can use the convergence guarantee of (5) to interpolate (“in edge weight space”) between an original graph G and its coarse grained version $\underline{G}$ via a family of intermediate graphs $\{G_\omega\}_\omega$ with ever increasing intra cluster weights.

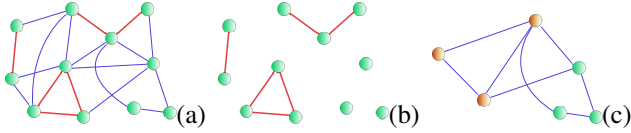


Figure 5. (a) G_ω . (b) Scaled edges of G_ω in red. (c) $\underline{G}$.

As we let $\omega \rightarrow \infty$, the heat kernel on G then more and more resembles the one on $\underline{G}$. To visualize this fact, we exemplarily plot in Fig. 6 (c) the difference $\|e^{-L_\omega t} - J^\uparrow e^{-\underline{L} t} J^\downarrow\|$ for the coarse graining setting of Figure 6 (a, b).

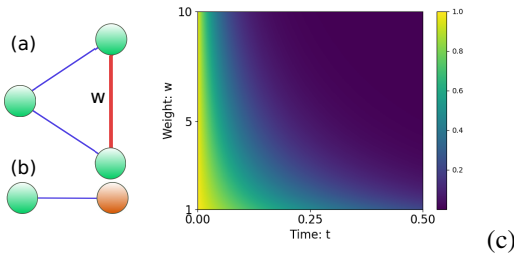


Figure 6. $\|e^{-L_\omega t} - J^\uparrow e^{-\underline{L} t} J^\downarrow\|$ -plot for graphs (a) & (b)

For fixed $t > 0$ we see that $\|e^{-L_\omega t} - J^\uparrow e^{-\underline{L} t} J^\downarrow\| \rightarrow 0$ as ω increases. Additionally, the decay $\|e^{-L_\omega t} - J^\uparrow e^{-\underline{L} t} J^\downarrow\| \rightarrow 0$ 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.

6.2. From Heat Kernels to Propagation Schemes

From (5) and Fig. 6 we conclude that applying the matrix e^{-tL_ω} on G is more and more the same as projecting to $\underline{G}$ via $J^\downarrow$, applying $e^{-t\underline{L}}$ there and interpolating back via $J^\uparrow$.

For a network to be continuous across scales, we need exactly this property: We want the propagation scheme on G_ω to more and more resemble the one on $\underline{G}$ as we increase

$\omega \rightarrow \infty$. As we have seen in Section 4, the propagation rule $X \mapsto \hat{A}XW$ of GCN does not achieve this, as it leads to disconnected limit graphs instead. However, propagating as

$$X \mapsto e^{-tL} XW, \quad (6)$$

does facilitate contact and similarity between information flows over G_ω and $\underline{G}$ as $\omega \rightarrow \infty$.

More generally, suppose we have for each time $t > 0$ individually that $\|e^{-L_\omega t} - J^\uparrow e^{-t\underline{L}} J^\downarrow\| < \delta$. If we build up the propagation matrix $\psi(L_\omega)$ as a weighted sum of such diffusion flows e^{-tL_ω} that have progressed to various times ($\psi(L_\omega) \sim \sum_k a_k e^{-t_k L_\omega}$) and the coefficients $\{a_k\}_k$ are not too large, then we can estimate $\|\psi(L_\omega) - J^\uparrow \psi(\underline{L}) J^\downarrow\| \leq (\sum_k |a_k|) \cdot \delta =: \epsilon$ by a triangle-inequality argument. Thus we can still guarantee that for large ω the propagation implemented by the layer-wise update rule

$$X \mapsto \psi(L_\omega) XW \quad (7)$$

over G_ω is approximately the same as the effective propagation $X \mapsto [J^\uparrow \psi(\underline{L}) J^\downarrow] XW$ over $\underline{G}$.

Generalizing the weighted sum to an integral, we define:

Definition 6.1. Let $\hat{\psi}$ be a bounded (generalized) function on $[0, \infty)$. The corresponding **Laplace-transform propagation matrix** is the matrix $\psi(L) \in \mathbb{R}^{N \times N}$ arising as the Laplace transform (cf. Appendix C for details) of $\hat{\psi}$:

$$\psi(L) := \int_0^\infty e^{-tL} \hat{\psi}(t) dt \quad (8)$$

Allowing *generalized* functions means we e.g. allow:

- Dirac distributions $\hat{\psi}_{\delta_s}(t) := \delta(t - s)$; yielding back **heat kernels** $\psi_s(L) = \int_0^\infty \delta(t - s) e^{-tL} dt = e^{-sL}$.
- Decaying functions $\hat{\psi}_k := (-t)^{k-1} e^{-\lambda t}$ yielding k^{th} -powers of **resolvents** $\psi_k(L) = [(\lambda I + L)^{-1}]^k$.

For such matrices $\Psi(L)$ we note (details in Appendix F.2):

Lemma 6.1 (informal). *Using Laplace-transform propagation matrices as in (8) together with the propagation rule (7) in each layer leads to scale continuous networks: If $G_\omega \rightarrow \underline{G}$ in the sense of heat kernels, then $F_\omega \rightarrow \underline{F}$ for the corresponding generated latent representations $F_\omega, \underline{F}$.*

7. The Result: Scale-Continuous GNNs

Lemma 6.1 above guarantees networks built on the propagation rule (7) are continuous across scales. We may think of this propagation rule as a scale continuous modification to the propagation rule (4) of GCN. GCN in turn is simultaneously both the simplest spectral- as well as message-passing

based graph neural network. Our next goal is therefore to leverage the insights obtained from making GCN scale continuous to extend scale continuity to general spectral- and message-passing based GNNs.

7.1. Laplace Transform Propagation

Spectral Networks: In order to turn spectral networks continuous across scales, we replace the polynomials $T_\ell(\mathcal{L})$ of the normalized Laplacian $\mathcal{L}$ used in their standard layer-wise update (3) with functions $\psi_\ell(L)$ of the graph Laplacian L as defined in (8). The resulting layer wise update reads

$$X^{(k+1)} = \sum_{\ell=0}^K \psi_\ell(L) X^{(k)} W_\ell. \quad (9)$$

Message Passing Networks: To determine which modifications are needed in order to turn message passing based networks scale continuous, we gain intuition from the the binary edge setting ($A_{uv} \in \{0, 1\}$): There we may write the definition of the message function in (1) equivalently simply as $m_{uv} = A_{uv} \cdot \phi(h_v^{(k)}, h_u^{(k)})$. As we saw in Sections 4 & 5, such adjacency-focused propagation schemes result in networks that are not continuous across scales. We hence make the replacement $A \mapsto \psi(L)$ in our propagation scheme, and choose sum as the aggregation function:

$$m_{vu}^{(k)} = [\psi(L)]_{vu} \cdot \phi(h_v^{(k)}, h_u^{(k)}) \quad (10)$$

$$h_v^{(k+1)} = \sum_{u \in G} m_{vu}. \quad (11)$$

Additionally one initial pass $h_0 = \Psi(L)X$ on the initial node features X is performed, before feeding them into our scale-continuous message passing framework (10) & (11).

Scalability: At first glance, propagation along Laplace-transform matrices as in (8) may appear difficult to scale, since matrices such as $\psi(L) = e^{-tL}$ are typically dense. A naïve implementation would thus require sending N^2 messages over an N -node graph. This does however not pose a practical bottleneck: most entries are vanishingly small ($[\psi(L)]_{ij} \ll \delta$), decaying exponentially with (graph- or resistance-) distance $d(i, j)$ between nodes i and j (Bauer et al., 2017). Hence, propagation remains effectively sparse, and negligible edges can be safely discarded in practice. We explore this further in Appendix I.

7.2. Theoretical Guarantees

Modifying spectral- and message-passing based graph neural networks exactly as outlined then indeed yields scale continuous networks, as we show next. Our main Theorem directly below establishes scale-continuity of the modified GNNs outlined in Section 7.1:

Theorem 7.1. *Let F and $\underline{F}$ be the latent embeddings generated for a graph G its coarsified version $\underline{G}$ by a (spectral or message passing) network employing Laplace transform propagation, as outlined in Section 7.1. With $\{\Psi_i(L) = \int_0^\infty \hat{\psi}_i(t) e^{-tL} dt\}_{i \in I}$ the collection of all Laplace-transform propagation matrices used within the network, we have $\|F - \underline{F}\| \leq C_\theta \cdot \max_i \left\{ \int_0^\infty |\hat{\psi}_i(t)| \cdot \|e^{-tL_\omega} - J^\dagger e^{-tL} J^\downarrow\| dt \right\}$. The constant C_θ depends on the learned parameters.*

The proof of this Theorem is provided in Appendix F.2. We note that whenever G_ω converges to $\underline{G}$ in the sense of heat kernels, by definition we have $\|e^{-tL_\omega} - J^\dagger e^{-tL} J^\downarrow\| \rightarrow 0$ for all $t > 0$. Hence the integrand in Theorem 7.1 converges to zero. Hence also $\|F_\omega - \underline{F}\|$ converges to zero.

Using the same proof strategy, one may also show:

Corollary 7.1. *In the setting of Theorem 7.1, for two graphs G_1, G_2 on the same node set, one has $\|F_1 - F_2\| \leq C_\theta \cdot \max_i \left\{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \right\}$.*

The proof is provided in Appendix E.2. Similar guarantees also hold at the node level, (cf. Appendix E.1).

8. Numerical Investigation of Scale Continuity

Continuity across scales: Theorem 7.1 implies that networks employing Laplace-transform propagation schemes are continuous as maps from the space of graphs into their latent spaces. To numerically verify this, we repeat the experiment of Section 4 for models belonging to this category (using resolvent and exponential matrices; cf. Definition 6.1 in a spectral and a spatial setting). As is evident from Fig. 7, latent embeddings generated by models employing Laplace transform propagation *do* indeed converge ($\|F_\omega - \underline{F}\| \rightarrow 0$). Hence for the models constructed in Section 7.1, we have indeed verified the desired continuity across scales.

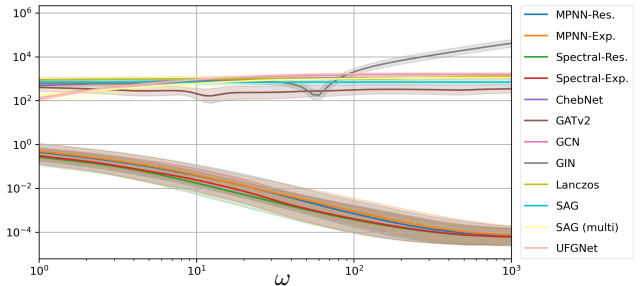


Figure 7. Latent distance $\|F_\omega - \underline{F}\|$. Latent embeddings generated by Laplace transform based GNNs converge; others do not.

Resulting Generalization Ability: In Section 4 we had identified lack of continuity as the obstruction to generalizing across scales. As verified above, graph neural networks based on Laplace-transform propagation *are* continuous. Hence we expect them to map similar graphs to similar

latent embeddings. To verify this, we here repeat the generalization experiment of Tables 1 & 2 in Section 3.

Table 3. Embedding difference $\|F - \underline{F}\|$ across resolution scales averaged over 5 runs (mean $\pm$ std). Lower is better ($\downarrow$).

Training $\rightarrow$ Inference:	High $\rightarrow$ Low	Low $\rightarrow$ High
GCN	20.3 $\pm$ 1.6	18.0 $\pm$ 0.6
GATv2	76.8 $\pm$ 52.4	42.5 $\pm$ 24.3
ChebNet	123.0 $\pm$ 3.8	158.5 $\pm$ 57.40
GIN	881.2 $\pm$ 268.7	2875.8 $\pm$ 1527.7
SAG	113.5 $\pm$ 10.3	43.3 $\pm$ 5.4
SAG-M	75.0 $\pm$ 15.1	61.7 $\pm$ 4.6
UFGNet	24.5 $\pm$ 0.3	26.8 $\pm$ 2.1
Lanczos	1286.1 $\pm$ 54.4	728.4 $\pm$ 230.0
MPNN _{exp.}	0.2 $\pm$ 0.1	0.2 $\pm$ 0.1
MPNN _{res.}	0.2 $\pm$ 0.1	0.2 $\pm$ 0.2
Spectral _{exp.}	0.3 $\pm$ 0.1	0.3 $\pm$ 0.1
Spectral _{res.}	0.3 $\pm$ 0.1	0.3 $\pm$ 0.3

Comparing with standard GNNs we see that in cross-resolution settings the difference $\|F - \underline{F}\|$ of latent embeddings generated by methods employing global Laplacian propagation schemes are *lower* than those of standard graph learning methods by factors of order 10^2 to 10^4 , indicating that the methods developed in Section 7.1 indeed *generalize*.

The small-to-negligible variations in latent representations then translate to small-to-negligible variations in prediction performance: As we infer from Table 4 below, MAEs generated by Laplace-Transform based GNNs in the cross resolution setting are essentially the same as those corresponding to same-resolution settings. Hence GNNs based on Laplace Transform propagation schemes are indeed able to *generalize* across scales.

Table 4. QM7 regression. Mean Absolute Error (MAE $\downarrow$) in kcal/mol for training and inference at different resolutions scales.

Resolution	MAE ($\downarrow$) on QM7 [kcal/mol]			
Training:	High		Low	
Inference:	Low	High	Low	High
GCN	136.7 $\pm$ 6.6	(63.6 $\pm$ 1.3)	(63.6 $\pm$ 1.3)	138.1 $\pm$ 2.4
GATv2	423.5 $\pm$ 337.1	(67.4 $\pm$ 8.2)	(59.7 $\pm$ 2.7)	257.4 $\pm$ 139.1
ChebNet	447.8 $\pm$ 6.0	(66.7 $\pm$ 1.4)	(71.5 $\pm$ 2.1)	158.7 $\pm$ 57.4
GIN	658.4 $\pm$ 85.8	(17.2 $\pm$ 2.8)	(38.3 $\pm$ 21.6)	1835.4 $\pm$ 925.8
SAG	589.7 $\pm$ 44.9	(68.2 $\pm$ 2.6)	(107.9 $\pm$ 1.1)	283.6 $\pm$ 39.3
SAG-M	194.4 $\pm$ 29.9	(66.6 $\pm$ 1.9)	(77.8 $\pm$ 6.0)	219.5 $\pm$ 11.7
UFGNet	131.5 $\pm$ 6.1	(62.4 $\pm$ 0.7)	(69.4 $\pm$ 0.7)	148.1 $\pm$ 6.3
Lanczos	938.4 $\pm$ 2.5	(9.9 $\pm$ 2.5)	(88.2 $\pm$ 2.7)	658.6 $\pm$ 199.2
MPNN _{exp.}	17.4 $\pm$ 3.4	(17.4 $\pm$ 3.4)	(17.9 $\pm$ 5.2)	17.9 $\pm$ 5.2
MPNN _{res.}	18.0 $\pm$ 4.3	(18.0 $\pm$ 4.3)	(17.6 $\pm$ 4.9)	17.6 $\pm$ 4.9
Spectral _{exp.}	15.9 $\pm$ 1.1	(15.9 $\pm$ 1.1)	(16.0 $\pm$ 1.5)	16.0 $\pm$ 1.5
Spectral _{res.}	17.2 $\pm$ 3.1	(17.2 $\pm$ 3.1)	(15.8 $\pm$ 1.4)	15.8 $\pm$ 1.4

Scale Continuity at the Node Level: We next consider popular citation networks (cf. Appendix G.4 for details) where nodes correspond to scientific articles. Labels correspond to the academic discipline of the paper and an edge implies a citation. We then expand individual nodes into connected k -cliques (cf. Fig. 8). We might interpret this as dissecting each article into subsections, which reference each other. Both typical models and the Laplace-transform propagation based methods introduced in Section 7.1 were then trained on the same (k -fold expanded) train-set and asked to classify nodes in the (k -fold expanded) test-partition.

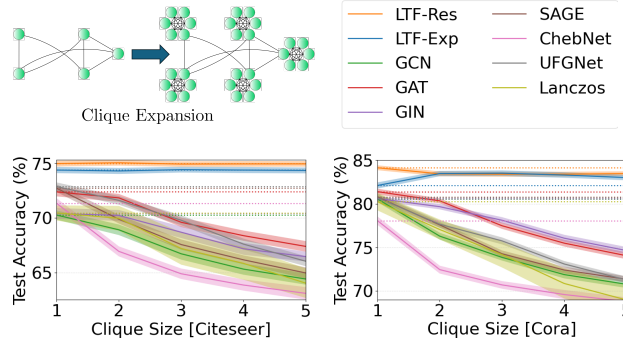


Figure 8. Node-Classification-Accuracy ($\uparrow$) and uncertainty (for 100 runs) vs. clique size.

The classification accuracies of methods not employing Laplace-transform propagation decrease significantly with increasing clique size (cf. Fig. 8). We can understand the underlying reason for this using GCN as an Example (cf. Appendix D for discussions on 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: GCNs propagation is more and more similar to a modified version of the original graph where edges *between* cliques are removed (G_2 in Fig. 9 (a) bottom).

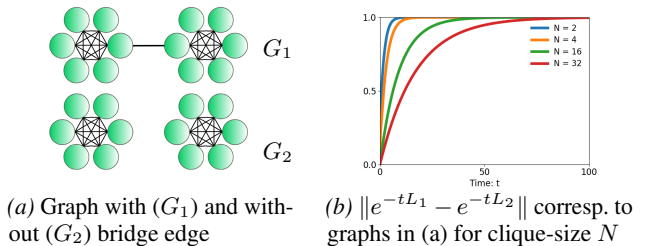


Figure 9. Graphs G_1, G_2 whose renormalized adjacency matrices are similar ($\hat{A}_1 \approx \hat{A}_2$), but whose heat kernels differ significantly.

This is not the case for Laplace-transform propagation based networks (using either resolvent or exponential propagation

matrices): As Corollary 7.1 elucidates, similarity of graphs for such models is not determined through the renormalized adjacency as for GCN. Rather it is governed by the similarity the heat kernels corresponding to the graphs, (i.e. the size of $\|e^{-tL_1} - e^{-tL_2}\|$ for any $t > 0$). As we can deduce from Fig. 9 (b) this quantity does not tend to zero with increasing clique size. Hence for the models of Section 7.1, propagation *does not* happen along a disconnected effective graph. Thus they are able to propagate information also *between* high connectivity areas and therefore are able to retain a high classification accuracy as the resolution scale is increased.

Scale continuity in the binary edge weight setting: In Section 6.1 we argued that from a diffusion perspective, as edge-weights within certain sub-graphs of a graph G tend to infinity, each such sub-graph effectively behaves as a single node in a coarse grained effective graph $\underline{G}$. This allows to continuously interpolate between G and its coarse grained version $\underline{G}$ by scaling up edge weights within clusters. In order to be able to generalize across scales, we then modified standard GNNs to turn them continuous with respect to this limiting procedure in Section 7.1.

We next test continuity of GNNs under a related but distinct graph-limiting procedure: Conceptually we can think of increasing the weights of edges within clusters as a proxy for increasing the *connectivity* within these subgraphs. To increase connectivity, rather than increasing the weights of a fixed *number* of edges, we might instead also increase the number of edges within a cluster. To do this, we make use of stochastic block models (c.f. Appendix G.3) (SBM) and vary the intra-cluster probability from $p = 0$ to $p = 1$.

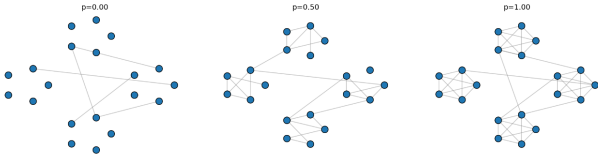


Figure 10. Graphs drawn from an SBM as the intra-cluster connectivity is varied from $p = 0$ to $p = 1$.

We then compare latent embeddings F_p generated for graphs drawn from an SBM at intra-cluster connectivity p with the latent embeddings $\underline{F}$ generated for a coarse grained version of this graph where clusters are aggregated to single nodes. As is evident from Fig. 11 below, latent embeddings generated by Laplace transform based models increasingly approximate the latent embedding $\underline{F}$ as $p \rightarrow 1$, while the same is not true for standard GNNs. In latent space, this continuity allows GNNs of Section 7.1 to embed clique graphs close to graphs where each such clique is represented as a supernode (cf. Table 5)).

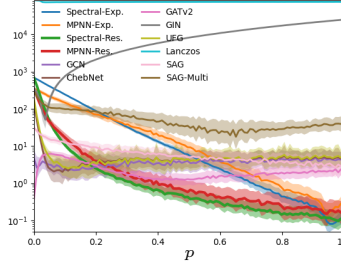


Figure 11. $\mathbb{E}[\|F_p - \underline{F}\|]$

Table 5. Latent distances

Model	$\ F_{p=1} - \underline{F}\ $
GCN	4.39 ± 2.95
GATv2	2.15 ± 1.02
ChebNet	4.67 ± 2.76
GIN	24877.22 ± 279.41
SAG	3.20 ± 2.51
SAG-M	43.46 ± 22.56
UFGNet	5.02 ± 3.75
Lanczos	74203.45 ± 576.27
Spectralres.	0.12 ± 0.07
Spectralexp.	0.13 ± 0.08
MPNNres.	0.22 ± 0.18
MPNNexp.	0.30 ± 0.17

Scale Continuity in the Continuum Limit: Finally, we consider the setting where graphs discretize a continuous manifold. In this regime, we require that as we increase the resolution scale, model outputs converge to those that would be produced by a model operating directly on the underlying continuous manifold. In particular we want the correlation between two nodes representing (the same two) points on the manifold to stabilize, as the resolution is increased.

To show that this requirement is not fulfilled by standard GNNs, we feed standard models with a dirac impulse at a fixed point (located at $\phi = 0, \theta = 0$) on the torus $\mathbb{T} = [0, 2\pi)^2$. We then investigate the output the GNNs generate at a test node, which is positioned at angles $\phi = \pi/2, \theta = 0$.

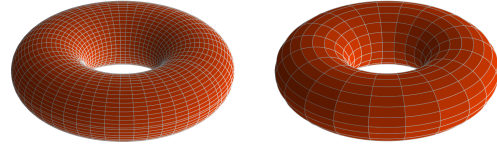


Figure 12. Graphs approximating torus-manifold (two resolutions).

As evident from Fig. 13 (a), this requirement is not fulfilled for standard GNNs, as as neither models' response stabilizes: For most GNNs the impuls response drops to zero, while for Lanczos it fluctuates as the mesh resolution (i.e. number of nodes) is increased.

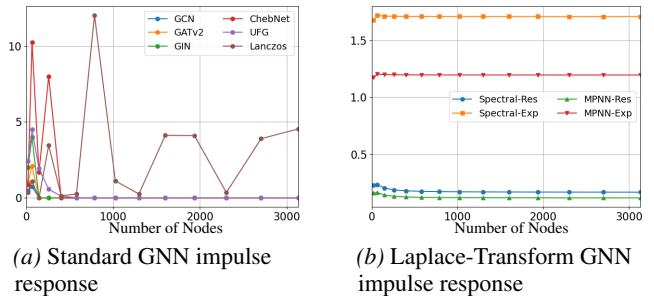


Figure 13. Impulse response

In contrast to that, the impulse response for the Laplace-transform based methods of Section 7.1 stays consistent as the mesh resolution is varied. To show that this persists

also at graph level, we would ideally want to show that the graph level latent embeddings F_N generated for a mesh on N nodes converge to the latent embedding corresponding to the continuous manifold as $N \rightarrow \infty$. In Appendix G.5 we detail that we indeed have actual convergence $F_N \rightarrow F$ towards a limit embedding F corresponding to the underlying continuous manifold. Since F is impossible to compute numerically, we verify the condition $\|F_N - F_{2N}\| \xrightarrow{N \rightarrow \infty} 0$ (necessary for convergence) in Fig. 14 below instead.

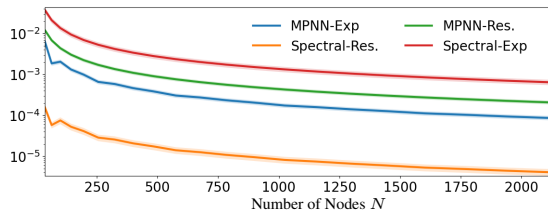


Figure 14. Graph level latent distance $\|F_N - F_{2N}\|$ as N increases.

9. Discussion

Our paper analyzed the discontinuity of existing GNNs across scales. We found the underlying obstruction to originate in commonly-used propagation schemes. We derived modifications to turn GNNs continuous, and showed that these modified models based on Laplace-Transform propagation can indeed *consistently* incorporate varying scales.

Acknowledgements

This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI). C.K. thanks Madeline Navarro for helpful discussions during the rebuttal phase. M.B. is partially supported by the EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub No. EP/Y028872/1.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

- Bauer, F., Hua, B., and Yau, S.-T. Sharp davies–gaffney–grigor’yan lemma on graphs. *Mathematische Annalen*, 368(3–4):1429–1437, 2017. doi: 10.1007/s00208-017-1529-z.
- Bianchi, F. M., Grattarola, D., Livi, L. F., and Alippi, C. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3496–3507, 2019.
- Blum, L. C. and Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.
- Brody, S., Alon, U., and Yahav, E. 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>.
- Bronstein, M. M., Bruna, J., Cohen, T., and Velicković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.
- Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLIS, April 2014*, 2014.
- Chung, F. R. K. *Spectral Graph Theory*. American Mathematical Society, 1997.
- Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.
- Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- Fey, M., Sunil, J., Nitta, A., Puri, R., Shah, M., Stojanović, B., Bendias, R., Barghi, A., Kocijan, V., Zhang, Z., He, X., Lenssen, J. E., and Leskovec, J. Pyg 2.0: Scalable learning on real world graphs. *arXiv preprint arXiv:2507.16991*, 2025. URL <https://arxiv.org/abs/2507.16991>.
- Gama, F., Ribeiro, A., and Bruna, J. 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>.
- Gama, F., Bruna, J., and Ribeiro, A. 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>.
- Gasteiger, J., Bojchevski, A., and Günnemann, S. 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, 2019. URL <https://openreview.net/forum?id=H1gL-2A9Ym>.

- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.
- Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (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/5dd9db5e033da9c6fb5ba83c7a7e9bea9-Abstract.html>.
- He, M., Wei, Z., Huang, Z., and Xu, H. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (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/76f1cfd7754a6e4fc3281bccb3d0902-Abstract.html>.
- He, M., Wei, Z., and Wen, J. Convolutional neural networks on graphs with chebyshev approximation, revisited. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/2f9b3ee2bcea04b327c09d7e3145bd1e-Abstract-Conference.html.
- Hein, M., Audibert, J.-Y., and von Luxburg, U. Graph laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 8:1325–1368, 2006. URL <https://api.semanticscholar.org/CorpusID:1355782>.
- Holland, P. W., Laskey, K. B., and Leinhardt, S. 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](https://doi.org/10.1016/0378-8733(83)90021-7). URL <https://www.sciencedirect.com/science/article/pii/0378873383900217>.
- Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.
- Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020. doi: 10.48550/arXiv.1905.12265. URL <https://arxiv.org/abs/1905.12265>.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zidek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-020-03085-1.
- Kato, T. *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>.
- Kenlay, H., Thanou, D., and Dong, X. On the stability of graph convolutional neural networks under edge rewiring. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021. URL https://openreview.net/forum?id=NG7Eb2_zR6K.
- Kipf, T. N. and Welling, M. 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>.
- Koke, C. 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>.
- Koke, C. Large coupling convergence beyond definiteness. *arXiv preprint arXiv:2601.18055*, 2026a. URL <https://arxiv.org/abs/2601.18055>.
- Koke, C. Di-graphs with tightly connected clusters: effective graph laplacians and resolvent convergence. *arXiv preprint arXiv:2601.18057*, 2026b. URL <https://arxiv.org/abs/2601.18057>.
- Koke, C. and Cremers, D. Holonets: Spectral convolutions do extend to directed graphs. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=EhmEwfavOW>.
- Koke, C. and Kutyniok, G. Graph scattering beyond wavelet shackles. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2020, November 28 - December 9, 2022, New Orleans, 2022*. URL <https://openreview.net/forum?id=ptUZl8xDMMN>.

- Koke, C., Saroha, A., Shen, Y., Eisenberger, M., and Cremers, D. Resolvnet: A graph convolutional network with multi-scale consistency. In *NeurIPS 2023 Workshop on New Frontiers in Graph Learning (GLFrontiers)*. OpenReview, October 2023. URL <https://openreview.net/forum?id=V5eDwEDfXT>.
- Koke, C., Saroha, A., Shen, Y., Eisenberger, M., Bronstein, M. M., and Cremers, D. Transferability for graph convolutional networks. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL <https://openreview.net/forum?id=rKEdfcaqYX>.
- Koke, C., Schnaus, D., Shen, Y., Saroha, A., Eisenberger, M., Rieck, B., Bronstein, M. M., and Cremers, D. On multi-scale graph representation learning. In *Learning Meaningful Representations of Life (LMRL) Workshop at ICLR 2025*, 2025a. URL <https://openreview.net/forum?id=X3OMHwfsxk>.
- Koke, C., Shen, Y., Saroha, A., Eisenberger, M., Rieck, B., Bronstein, M. M., and Cremers, D. Graph networks struggle with variable scale. In *ICLR 2025 Workshop ICBINB*. OpenReview, March 2025b. URL <https://openreview.net/forum?id=N5n6SAfnU0>.
- Koke, C., Shen, Y., Saroha, A., Eisenberger, M., Rieck, B., Bronstein, M. M., and Cremers, D. On incorporating scale into graph networks. In *ICLR 2025 Workshop on Machine Learning for Multiscale Physics (MLMP)*. OpenReview, March 2025c. URL <https://openreview.net/forum?id=SRCsyJafgP>.
- Koke, C., Rieck, B., Bronstein, M. M., and Cremers, D. Scale continuity in graph learning: Going beyond spectral methods. In *ICLR 2026 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2026. URL <https://openreview.net/forum?id=RYl2VWfIRI>.
- Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Meroze, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., and Battaglia, P. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336. URL <https://www.science.org/doi/abs/10.1126/science.adi2336>.
- Le, T. and Jegelka, S. 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>.
- Lee, J., Lee, I., and Kang, J. Self-attention graph pooling. In Chaudhuri, K. and Salakhutdinov, R. (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>.
- Levie, R., Bronstein, M. M., and Kutyniok, G. Transferability of spectral graph convolutional neural networks. *CoRR*, abs/1907.12972, 2019a. URL <http://arxiv.org/abs/1907.12972>.
- Levie, R., Isufi, E., and Kutyniok, G. On the transferability of spectral graph filters. *CoRR*, abs/1901.10524, 2019b. URL <http://arxiv.org/abs/1901.10524>.
- Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. 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>.
- Liao, R., Zhao, Z., Urtasun, R., and Zemel, R. S. 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>.
- Loukas, A. 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>.
- Loukas, A. and Vandergheynst, P. Spectrally approximating large graphs with smaller graphs. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3243–3252. PMLR, 2018a. URL <http://proceedings.mlr.press/v80/loukas18a.html>.
- Loukas, A. and Vandergheynst, P. Spectrally approximating large graphs with smaller graphs. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3237–3246. PMLR, 10–15 Jul 2018b. URL <https://proceedings.mlr.press/v80/loukas18a.html>.
- Mallat, S. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10): 1331–1398, 2012. doi: <https://doi.org/10.1002/cpa>.

21413. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21413>.
- Maskey, S., Levie, R., and Kutyniok, G. Transferability of graph neural networks: an extended graphon approach. *CoRR*, abs/2109.10096, 2021. URL <https://arxiv.org/abs/2109.10096>.
- McCallum, A., Nigam, K., Rennie, J., and Seymore, K. 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>.
- Perlmutter, M., Gao, F., Wolf, G., and Hirn, M. Understanding graph neural networks with asymmetric geometric scattering transforms, 2019. URL <https://arxiv.org/abs/1911.06253>.
- Post, O. *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.
- Price, C. C., Singh, A., Frey, N. C., and Shenoy, V. B. Efficient catalyst screening using graph neural networks to predict strain effects on adsorption energy. *Science Advances*, 8(47):eabq5944, 2022. doi: 10.1126/sciadv.abq5944. URL <https://www.science.org/doi/abs/10.1126/sciadv.abq5944>.
- Roddenberry, T. M., Gama, F., Baraniuk, R. G., and Segarra, S. 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>.
- Ruiz, L., Chamon, L. F. O., and Ribeiro, A. Graphon neural networks and the transferability of graph neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (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/12bcd658ef0a540cab36cdf2b1046fd-Abstract.html>.
- Rupp, M., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108:058301, 2012.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. 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>.
- Tao, T. *An Introduction to Measure Theory*. Graduate studies in mathematics. American Mathematical Society, 2013. ISBN 9781470409227. URL <https://books.google.de/books?id=SPGJjwEACAAJ>.
- Teschl, G. *Mathematical Methods in Quantum Mechanics*. American Mathematical Society, 2014.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. 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>.
- Wang, Z., Ruiz, L., and Ribeiro, A. 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>.
- Wang, Z., Ruiz, L., and Ribeiro, A. 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>.
- Wang, Z., Ruiz, L., and Ribeiro, A. 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>.
- Wihler, T. 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.
- Xie, T. and Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14): 145301, 2018. doi: 10.1103/PhysRevLett.120.145301.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. 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>.

Zheng, X., Zhou, B., Gao, J., Wang, Y., Lió, P., Li, M., and Montúfar, G. How framelets enhance graph neural networks. In Meila, M. and Zhang, T. (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>.

Zou, D. and Lerman, G. 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. Discussion of Tasks and Graph Structures for which Scale Continuity is important

When GNNs are tasked to strictly distinguish non-isomorphic graphs the exact articulation of a graph matters (e.g. program graphs, electrical circuit graphs, etc.). In other settings, multiple graphs may describe the same underlying object (graphs discretizing the same manifold, sampled from the same distribution, or describing the same underlying noisy (social-, or natural-) science dataset. Here we want models to effectively embed the underlying object and not overfit on exact fine-print articulation of graphs describing said object.

A notion of distance respecting this intuition arises from comparing heat kernels as $\|e^{-tL_1} - e^{-tL_2}\|$. Using exponentials leads to an (exponential) suppression of large eigenvalues of the Laplacian L . The suppressed information encoded into these large eigenvalues corresponds to fine structure details of the graph (c.f. e.g. (Chung, 1997)) which the model is supposed to discount. When comparing graphs on different numbers of nodes, we have to align them via projection $J^\downarrow$ and interpolation $J^\uparrow$ operators determining how to traverse between these graphs.

The quantity $\|e^{-tL_1} - J^\uparrow e^{-tL_2} J^\downarrow\|$ (c.f. e.g. eq. (5) & Fig. 6) measures similarity of coarse structures in G_1, G_2 , after alignment. If this difference is small, coarse structures are both aligned and similar. Only then should models generate similar embeddings.

B. Discussion of Related Work

In this section we provide further details on existing approaches to estimating variations in latent embeddings generated by 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 ($|\mathcal{E}|$ 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 illuminates 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. Since this approach is applicable only when graphs are large and $|\mathcal{E}| = \mathcal{O}(N^2)$, it e.g. does not cover the setting of Table 1.

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. Similar limitations as for (Ruiz et al., 2020) apply.

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 ($|\mathcal{E}| = \mathcal{O}(N)$) making use of the concept of Graphops, mathematical operators that can be used to model how GNNs behave on large sparse graphs. The authors also 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. Due to the necessity for graphs to be large (with edges $|\mathcal{E}| = \mathcal{O}(N)$) it also does not apply e.g. to the setting considered in Section 3.

On Local Distributions in Graph Signal Processing (Roddenberry et al., 2022): This work is rooted in the field of graph signal processing (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. 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. This work similarly is only applicable to large sparse graphs.

Graph Convolutional Neural Networks via Scattering (Zou & Lerman, 2020) This work provides a different perspective on spectral GNNs by connecting them to scattering transforms (Mallat, 2012; Perlmutter et al., 2019; Koke & Kutyniok, 2022), a concept from signal processing. The authors demonstrate that spectral GNNs 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. A key focus of the paper is the stability of networks when viewed through the scattering framework. The authors argue that scattering transforms offer a more stable approach to graph signal processing compared to traditional networks, especially in the presence of noisy or incomplete graph data. Derived single filter transferability results depend on spectral properties of the utilized Laplacians on the respective graphs. The focus of this work is on the setting where $\|L - \tilde{L}\|$ is small. This is distinct from the setting of similar heat kernels, which is more general.

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). Similarly to the filters analyzed in the present work, manifold filters are defined in terms of Laplace transforms. By framing spectral graph neural networks 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 technically, filters are bounded as $|\psi(L) - \psi(\tilde{L})| \leq K\|L - \tilde{L}\|$. 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). In contrast, our work does not primarily focus on (large) graphs approximating a common manifold. We only consider this setting as *one* example. In comparison to (Wang et al., 2024a), the results we develop for this particular example (c.f. Appendix G.5) need no assumptions on band-limitedness. Additionally our results also apply to message passing based networks, in contrast to those of (Wang et al., 2024a) which apply to spectral networks.

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 ($\|LP\phi - \mathcal{L}P\phi\|$) is considered. 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 (cf. 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 when applied to the specific example of discretized manifolds (cf. Appendix G.5) 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. Additionally, our results also cover message passing based networks, and are not restricted to spectral networks.

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. The study also develops sufficient conditions for achieving low transferability errors, demonstrating that spectral gnnns 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 (cf. Theorem 17 *ibid.*). However, signals are assumed to be bandlimited. In the specific example setting of graphs discretizing manifolds, we avoid Levie’s growth of the stability constant with the number of considered eigenvalues (cf. 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. Additionally, our results also cover message passing based networks, and are applicable also outside the setting of graphs discretizing ambient spaces.

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. 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. Results derived in this work do not apply to our settings, as detecting graph variations via the difference $\|\mathcal{L} - \tilde{\mathcal{L}}\|$ as considered in (Gama et al., 2019) is insensitive to the notion of variation and convergence we consider here (c.f. Section 6.1). Additionally the use of the normalized Laplacian in (Gama et al., 2019) precludes detection of the convergence behaviour of Section 6.1.

Stability Properties of Graph Neural Networks (Gama et al., 2020): This paper investigates the stability properties of 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 spectral 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. Results derived in this work do not apply to our settings, as detecting graph variations via the difference $\|L - \tilde{L}\|$ is insensitive to the notion of variation and convergence we consider here (c.f. Section 6.1).

Limitless transferability for graph convolutional Networks (Koke, 2023): This work studies stability- and transferability properties 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 spectral filter functions which constitute an example of the more general class of Laplace transform filters considered in this present work. Our work here is however far more general, considering general Laplace transform based propagations schemes, including not only spectral-, but also message passing based networks. Additionally we develop a rich and detailed theory and provide numerical investigations and experimental evaluations in diverse settings.

Previous versions of this present work: Earlier versions and preliminary aspects of this research were presented at various non-archival workshops (Koke et al., 2023; 2024; 2025a;b;c; 2026). The various aspects discussed there are subsumed and extended by this present work.

C. Laplace Transform Matrices

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

C.1. Complex measures on $\mathbb{R}_{\geq 0}$ 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

$$\mathbb{R}_{\geq 0} = [0, \infty). \quad (12)$$

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

- $\emptyset, \mathbb{R}_{\geq 0} \in \Sigma$
- $A, B \in \sigma \Rightarrow A \cap B \in \Sigma$
- $A, B \in \Sigma \Rightarrow A \setminus B \in \Sigma$
- $A, B \in \Sigma \Rightarrow A \cup B \in \Sigma$.

We now take $\Sigma_{\mathbb{R}_{\geq 0}}$ 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 $\Sigma_{\mathbb{R}_{\geq 0}}$ a complex number in a certain way:

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

$$\mu \left(\bigcup_n A_n \right) = \sum_n \mu(A_n) \quad (13)$$

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

Let us provide some examples:

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

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

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

$$\chi_A(t) = \begin{cases} 1 & ; t \in A \\ 0 & ; t \notin A \end{cases} \quad (14)$$

as

$$f(t) \approx f_n(t) := \sum_k a_k^n \chi_{A_k}(t). \quad (15)$$

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

$$\int_{\mathbb{R}_{\geq 0}} f_n d\mu \equiv \sum_k a_k^n \cdot \mu(A_k). \quad (16)$$

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

$$\int_{\mathbb{R}_{\geq 0}} f d\mu \equiv \lim_{n \rightarrow \infty} \int_{\mathbb{R}_{\geq 0}} f_n d\mu. \quad (17)$$

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

$$\int_{\mathbb{R}_{\geq 0}} f(t) d\mu_{\text{Leb}}(t) = \int_0^\infty f(t) dt. \quad (18)$$

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

$$\int_{\mathbb{R}_{\geq 0}} f(t) d\mu_{\delta_{t_0}}(t) = f(t_0) \quad (19)$$

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

$$\int_{\mathbb{R}_{\geq 0}} f(t) d\mu_{\hat{\psi}} = \int_0^\infty \hat{\psi}(t) f(t) dt. \quad (20)$$

C.2. Laplace Transforms

We say complex valued measure μ is finite if we have

$$\int_{\mathbb{R}_{\geq 0}} d|\mu|(t) < \infty. \quad (21)$$

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

$$|\mu|((a, b)) \equiv |\mu((a, b))|. \quad (22)$$

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

$$\psi_\mu(z) := \int_{\mathbb{R}_{\geq 0}} e^{-tz} d\mu(t). \quad (23)$$

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

$$\mathbb{C}_R := \{z \in \mathbb{C} : \operatorname{Re}(z) \geq 0\}. \quad (24)$$

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

$$|\psi_\mu(z)| = \left| \int_{\mathbb{R}_{\geq 0}} e^{-tz} d\mu(t) \right| \quad (25)$$

$$\leq \int_{\mathbb{R}_{\geq 0}} |e^{-tz}| d|\mu|(t) \quad (26)$$

$$\leq \int_{\mathbb{R}_{\geq 0}} d|\mu|(t) < \infty. \quad (27)$$

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

$$\psi_{\mu_{\delta_{t_0}}}(z) = e^{-t_0 z}. \quad (28)$$

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

$$\psi(z) \equiv \int_{\mathbb{R}_{\geq 0}} e^{-tz} d\mu_{\hat{\psi}} = \int_0^\infty \hat{\psi}(t) e^{-tz} dt. \quad (29)$$

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

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

$$\psi_k(z) = (z + \lambda)^{-k}. \quad (30)$$

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

$$\int_0^\infty e^{-tz} e^{-\lambda t} dt = -\frac{1}{z + \lambda} e^{-(z+\lambda)t} \Big|_0^\infty. \quad (31)$$

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 $\operatorname{Re}(z) \leq 0$, as long as $z \neq -\lambda$.

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

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

$$h(x) = \sum_k \theta_k \psi_k(x) \quad (32)$$

for which

$$\sup_{x \in [0, \infty)} |f(x) - h(x)| < \epsilon. \quad (33)$$

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. $\square$

D. 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 5. 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:

$$\mathcal{E}_{\text{high}} := \{(i, j) \in \mathcal{E} : w_{ij} \geq S_{\text{high}}\} \quad (34)$$

and the set with small weights is given by:

$$\mathcal{E}_{\text{reg.}} := \{(i, j) \in \mathcal{E} : w_{ij} \leq S_{\text{reg.}}\} \quad (35)$$

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. $S_{\text{high}} = \min_{(i,j) \in \mathcal{E}_{\text{high}}} w_{ij}$) and no weights in between the scales.

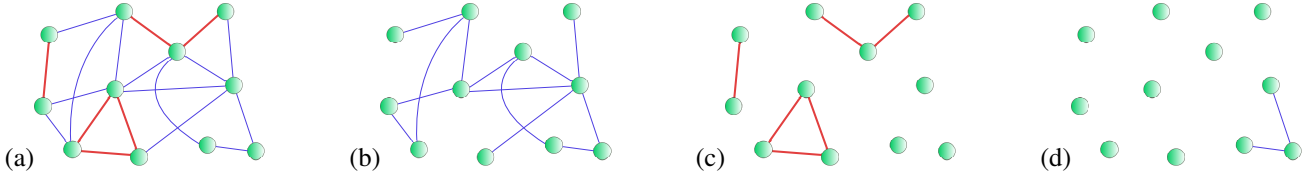


Figure 15. (a) Graph G with $\mathcal{E}_{\text{reg.}}$ (blue) & $\mathcal{E}_{\text{high}}$ (red); (b) $G_{\text{reg.}}$; (c) G_{high} ; (d) $G_{\text{reg., exclusive}}$

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. 15).

We also introduce the set of edges $\mathcal{E}_{\text{reg., exclusive}} := \{(i, j) \in \mathcal{E}_{\text{reg.}} \mid \forall k \in \mathcal{G} : (i, k) \notin \mathcal{E}_{\text{high}} \text{ \& } (k, j) \notin \mathcal{E}_{\text{high}}\}$ 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. 15 (d).

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

$$S_{\text{high}} \gg S_{\text{reg.}} \quad (36)$$

D.1. Spectral Convolutional Filters

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

$$x \mapsto g_{\theta}(T)x \quad (37)$$

for a node feature x , a learnable function g_{θ} and a graph shift operator T .

D.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

$$\|T\| \gg 1. \quad (38)$$

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

$$\sigma(g_\theta(T)) = \{g_\theta(\lambda) : \lambda \in \sigma(T)\}, \quad (39)$$

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

$$|\lambda_{\max}| = \|T\|. \quad (40)$$

Since learned functions are implemented directly as a polynomial (as e.g. in (Defferrard et al., 2016)), we have

$$\lim_{\lambda \rightarrow \pm\infty} |g_\theta(\lambda)| = \infty. \quad (41)$$

Thus in view of (39) and (40) we have for $\|T\|$ sufficiently large, that

$$\|g_\theta(T)\| = |g_\theta(\pm\|T\|)| \quad (42)$$

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

$$C \cdot \|T\| \leq \|g_\theta(T)\| \quad (43)$$

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

$$\|x\| \cdot C \cdot \|T\| \leq \|g_\theta(T)x\| \quad (44)$$

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 .

D.1.2. SPECTRAL NORMALIZATIONS

As discussed in the previous Section D.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

$$T = \frac{1}{\lambda_{\max}(\Delta)} \Delta, \quad (45)$$

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. 16) only, as Theorem D.1 below establishes:

Theorem D.1. With

$$T = \frac{1}{\lambda_{\max}(\Delta)} \Delta, \quad (46)$$

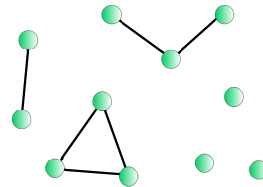


Figure 16. Limit graph corresponding to Fig 15 for spectral normalization

and the scale decomposition as above we have that

$$\left\| T - \frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \Delta_{\text{high}} \right\| = \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right) \quad (47)$$

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

Proof. For convenience in notation, let us write

$$T_{\text{high}} = \frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \Delta_{\text{high}} \quad (48)$$

and similarly

$$T_{\text{reg.}} = \frac{1}{\lambda_{\max}(\Delta_{\text{reg.}})} \Delta_{\text{reg.}}. \quad (49)$$

We may write

$$\Delta = \Delta_{\text{high}} + \Delta_{\text{reg.}}, \quad (50)$$

which we may rewrite as

$$\Delta = \lambda_{\max}(\Delta_{\text{high}}) \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}} \right). \quad (51)$$

Let us consider the equivalent expression

$$\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta = T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}}. \quad (52)$$

We next note that

$$\lambda_{\max} \left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta \right) = \frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})}. \quad (53)$$

and

$$\lambda_{\max}(T_{\text{high}}) = 1 \quad (54)$$

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

$$\lambda_{\max}(|a| \cdot T) = |a| \cdot \lambda_{\max}(T) \quad (55)$$

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

Since the right-hand-side of (52) 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

$$\lambda_{\max} \left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta \right) = 1 + \mathcal{O} \left(\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \right). \quad (56)$$

Using (53) and the fact that

$$\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}}, \quad (57)$$

we thus have

$$\frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})} = 1 + \mathcal{O} \left(\frac{S_{\text{reg.}}}{S_{\text{high}}} \right). \quad (58)$$

Since for small ϵ , we also have

$$\frac{1}{1 + \epsilon} = 1 + \mathcal{O}(\epsilon), \quad (59)$$

the relation (57) also implies

$$\frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}(\Delta)} = 1 + \mathcal{O} \left(\frac{S_{\text{reg.}}}{S_{\text{high}}} \right). \quad (60)$$

Multiplying (51) with $1/\lambda_{\max}(\Delta)$ yields

$$T = \frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}(\Delta)} \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}} \right). \quad (61)$$

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

$$\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}} < 1 \quad (62)$$

for sufficiently large S_{high} , relation (61) implies

$$\left\| T - \frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \Delta_{\text{high}} \right\| = \mathcal{O} \left(\frac{S_{\text{reg.}}}{S_{\text{high}}} \right) \quad (63)$$

as desired.

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

$$|\lambda_k(A) - \lambda_k(B)| \leq \|A - B\| \quad (64)$$

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 (56) that explicitly

$$\left| 1 - \frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})} \right| \leq \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})}. \quad (65)$$

This in turn can then be used to provide a quantitative bound in (47). Since we are only interested in the qualitative behaviour for $S_{\text{high}} \gg S_{\text{reg.}}$, we shall however not pursue this further. $\square$

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

Lemma D.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:

$$|\lambda_k(A) - \lambda_k(B)| \leq \|A - B\|. \quad (66)$$

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

$$|\lambda_i(A + B) - \lambda_i(A)| \leq \|B\| \quad (67)$$

for Hermitian A, B . Since we have

$$\lambda_i(A) - \lambda_i(A + B) = \lambda_i((A + B) + (-B)) - \lambda_i(A + B) \quad (68)$$

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

$$\lambda_i(A + B) - \lambda_i(A) \leq \|B\| \quad (69)$$

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

$$\lambda_i(M) = \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^* M v. \quad (70)$$

With this we find

$$\lambda_i(A+B) - \lambda_i(A) = \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*(A+B)v - \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Av \quad (71)$$

$$\leq \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Av + \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Bv \quad (72)$$

$$- \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Av \quad (73)$$

$$= \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Bv \quad (74)$$

$$= \sup_{\dim(V)=i} \inf_{v \in V, \|v\|=1} v^*Bv \quad (75)$$

$$\leq \max_{1 \leq k \leq n} \{|\lambda_k(B)|\} \quad (76)$$

$$= \|B\|. \quad (77)$$

□

D.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

$$\mathcal{L} = Id - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}. \quad (78)$$

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. 17) only, as Theorem D.3 below establishes:

Theorem D.3. With

$$T = Id - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \quad (79)$$

and the scale decomposition as introduced above, we have that

$$\left\| T - \left(Id - D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} - D_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}} \right) \right\| = \mathcal{O} \left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}} \right) \quad (80)$$

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

Proof. We first note that instead of (80), we may equivalently establish

$$\left\| D^{-\frac{1}{2}}WD^{-\frac{1}{2}} - \left(D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}} \right) \right\| = \mathcal{O} \left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}} \right). \quad (81)$$

We have

$$W = W_{\text{high}} + W_{\text{reg.}}. \quad (82)$$

With this, we may write

$$D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}} + D^{-\frac{1}{2}}W_{\text{reg.}}D^{-\frac{1}{2}}. \quad (83)$$

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

$$\left(D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j}} \quad (84)$$

Let us use the notation

$$d_i^{\text{high}} = \sum_{j=1}^N (W_{\text{high}})_{ij}, \quad d_i^{\text{reg.}} = \sum_{j=1}^N (W_{\text{reg.}})_{ij} \quad \text{and} \quad d_i^{\text{low, exclusive}} = \sum_{j=1}^N (W_{\text{low, exclusive}})_{ij}. \quad (85)$$

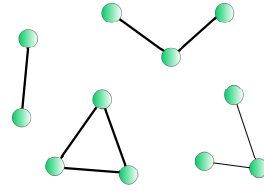


Figure 17. Limit graph corresponding to Fig 15 for symmetric normalization

We then find

$$\frac{1}{\sqrt{d_i}} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}} \quad (86)$$

Using the Taylor expansion

$$\frac{1}{\sqrt{1 + \epsilon}} = 1 - \frac{1}{2}\epsilon + \mathcal{O}(\epsilon^2), \quad (87)$$

we thus have

$$\left(D^{-\frac{1}{2}} W_{\text{high}} D^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} + \mathcal{O}\left(\frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}\right). \quad (88)$$

Since we have

$$\frac{d_i^{\text{reg.}}}{d_i^{\text{high}}} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}}, \quad (89)$$

this yields

$$D^{-\frac{1}{2}} W_{\text{high}} D^{-\frac{1}{2}} = D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right). \quad (90)$$

Thus let us turn towards the second summand on the right-hand-side of (83). We have

$$\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{d_i}} \cdot (W_{\text{reg.}})_{ij} \cdot \frac{1}{\sqrt{d_j}}. \quad (91)$$

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

$$\frac{1}{\sqrt{d_j}} = \frac{1}{\sqrt{d_j^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}}. \quad (92)$$

Since

$$\frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}}}{d_i^{\text{high}}}}} \leq 1, \quad (93)$$

we have

$$\left|\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij}\right| \leq \left|\frac{1}{\sqrt{d_i}} \cdot (W_{\text{reg.}})_{ij}\right| \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right). \quad (94)$$

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

$$\left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}}\right)_{ij} = \left(D_{\text{reg.}}^{-\frac{1}{2}} W_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}}\right)_{ij}. \quad (95)$$

Thus in total we have established

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = \left(D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}} W_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}}\right) + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right) \quad (96)$$

which was to be established. $\square$

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

$$T = \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}}, \quad (97)$$

with

$$\tilde{W} = (W + Id) \quad (98)$$

and

$$\tilde{D} = D + Id. \quad (99)$$

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

Theorem D.4. With

$$T = \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}}, \quad (100)$$

where $\tilde{W} = (W + Id)$ and $\tilde{D} = D + Id$ as well as the scale decomposition introduced above, we have that

$$\left\| T - \left(D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}} \tilde{W}_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}} \right) \right\| = \mathcal{O} \left(\sqrt{\frac{S_{\text{reg.}} + 1}{S_{\text{high}}}} \right) \quad (101)$$

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

$$\tilde{W}_{\text{low, exclusive}} := W_{\text{low, exclusive}} + \text{diag}(\mathbb{1}_{G_{\text{low, exclusive}}}) \quad (102)$$

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 D.3 is thus that applicability of the limit propagation scheme of Fig. 17 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 D.3 – we first decompose T :

$$\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}} = \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} Id \tilde{D}^{-\frac{1}{2}} \quad (103)$$

$$= \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-1} \quad (104)$$

For the first term, we note

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i + 1}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j + 1}}. \quad (105)$$

We then find

$$\frac{1}{\sqrt{d_i + 1}} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}} + 1}{d_i^{\text{high}}}}}. \quad (106)$$

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

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i^{\text{high}}}} \cdot (W_{\text{high}})_{ij} \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} + \mathcal{O} \left(\frac{1 + d_i^{\text{reg.}}}{d_i^{\text{high}}} \right). \quad (107)$$

This implies

$$\tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} = D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + \mathcal{O} \left(\frac{S_{\text{reg.}} + 1}{S_{\text{high}}} \right). \quad (108)$$

Next we turn to the second summand in (103):

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \frac{1}{\sqrt{d_i + 1}} \cdot (W_{\text{reg.}})_{ij} \cdot \frac{1}{\sqrt{d_j + 1}}. \quad (109)$$

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

$$\frac{1}{\sqrt{d_j + 1}} = \frac{1}{\sqrt{d_j^{\text{high}}}} \cdot \frac{1}{\sqrt{1 + \frac{d_j^{\text{reg.}} + 1}{d_j^{\text{high}}}}}. \quad (110)$$

Since

$$\frac{1}{\sqrt{1 + \frac{d_i^{\text{reg.}} + 1}{d_i^{\text{high}}}}} \leq 1, \quad (111)$$

we have

$$\left| \left(D^{-\frac{1}{2}} W_{\text{reg.}} D^{-\frac{1}{2}} \right)_{ij} \right| \leq \left| \frac{1}{\sqrt{1 + d_i}} \cdot (W_{\text{reg.}})_{ij} \right| \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} \quad (112)$$

$$\leq \left| \frac{1}{\sqrt{d_i^{\text{reg.}}}} \cdot (W_{\text{reg.}})_{ij} \right| \cdot \frac{1}{\sqrt{d_j^{\text{high}}}} \quad (113)$$

$$= \mathcal{O} \left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}} \right). \quad (114)$$

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

$$\left(\tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} \right)_{ij} = \left(\tilde{D}_{\text{reg.}}^{-\frac{1}{2}} W_{\text{low, exclusive}} \tilde{D}_{\text{reg.}}^{-\frac{1}{2}} \right)_{ij}. \quad (115)$$

Finally we note for the third term on the right-hand-side of (103) that

$$\frac{1}{d_i} \leq \frac{1}{d_i^{\text{high}}} = \mathcal{O} \left(\frac{1}{S_{\text{high}}} \right) \quad (116)$$

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

In total we thus have found

$$\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}} = \left(D_{\text{high}}^{-\frac{1}{2}} W_{\text{high}} D_{\text{high}}^{-\frac{1}{2}} + D_{\text{reg.}}^{-\frac{1}{2}} \tilde{W}_{\text{low, exclusive}} D_{\text{reg.}}^{-\frac{1}{2}} \right) + \mathcal{O} \left(\sqrt{\frac{S_{\text{reg.}} + 1}{S_{\text{high}}}} \right); \quad (117)$$

which was to be proved. $\square$

D.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))

$$X_i^{\ell+1} = \gamma \left(X_i^\ell, \coprod_{j \in \mathcal{N}(i)} \phi(X_i^\ell, X_j^\ell, w_{ij}) \right). \quad (118)$$

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 of Figure 4 (b) and are thus not stable to scale transitions and topological perturbations.

D.2.1. SCALE-SENSITIVITY OF MESSAGE PASSING NEURAL NETWORKS

Here we establish that message passing networks (as defined in (118) above) are unable to emulate a limit propagation scheme similar to the one in Figure 4 (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. 18). We assume all node-weights to be equal to one ($\mu_i = 1$ for $i = 1, 2, 3$) and edge weights

$$w_{13}, w_{23} \leq S_{\text{reg}}. \quad (119)$$

as well as

$$w_{12} = S_{\text{high}}. \quad (120)$$

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

Given states $\{X_1^\ell, X_2^\ell, X_3^\ell\}$ in layer ℓ , a limit propagation scheme as in Figure 4 (b) would require the updated feature vector of node 3 to be given by

$$X_{3,\text{desired}}^{\ell+1} := \gamma \left(X_3^\ell, \phi \left(X_3^\ell, \frac{X_1^\ell + X_2^\ell}{2}, (w_{31} + w_{32}) \right) \right) \quad (121)$$

However, the actual updated feature at node 3 is given as (c.f. (118)):

$$X_{3,\text{actual}}^{\ell+1} := \gamma \left(X_3^\ell, \phi \left(X_3^\ell, X_1^\ell, w_{31} \right) \amalg \phi \left(X_3^\ell, X_2^\ell, w_{32} \right) \right) \quad (122)$$

Since there is no dependence on S_{high} in equation (122) – 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 $X_{3,\text{actual}}^{\ell+1} \not\rightarrow X_{3,\text{desired}}^{\ell+1}$ does not converge as S_{high} increases.

D.2.2. LIMIT PROPAGATION SCHEMES

The number of possible choices of message functions ϕ , aggregation functions $\amalg$ 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 (122) 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

$$\phi(X_i^\ell, X_j^\ell, w_{ij}) = \text{ReLU} \left(W_1^\ell \cdot X_i^\ell + W_2^\ell \cdot X_j^\ell + W_3^\ell \cdot w_{ij} + B^\ell \right) \quad (123)$$

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-vector $W_3^{\ell+1}$ 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 distribution 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 $(W_3^\ell)_a$ of the weight vector $W_3^\ell \in \mathbb{R}^{F_{\ell+1}}$:

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

$$\lim_{w_{ij} \rightarrow \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = 0 \quad (124)$$

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

$$\lim_{w_{ij} \rightarrow \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \infty \quad (125)$$

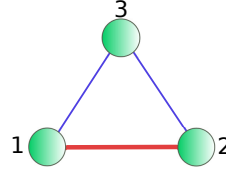


Figure 18. Three node Graph G with on large weight $w_{12} \gg 1$.

For either choice of aggregation function $\mathbb{I}$ in (118) among "max", "sum" or "mean" the behaviour in (125) 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 (125) 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 $(W_3^\ell)_a > 0$ dominate over messages propagated along edges in $\mathcal{E}_{\text{reg.}}$. By (124), 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 $(W_3^\ell)_a < 0$. 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 $(W_3^\ell)_a > 0$.

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 (124) and (125) are given as follows:

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

$$\lim_{w_{ij} \rightarrow \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \lim_{y \rightarrow -\infty} \sigma(y). \quad (126)$$

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

$$\lim_{w_{ij} \rightarrow \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \lim_{y \rightarrow \infty} \sigma(y). \quad (127)$$

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

$$X_i^{\ell+1} = \rho \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} (W X_j^{\ell+1}) \right). \quad (128)$$

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

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(\vec{a}^\top [W X_i^\ell \parallel W X_j^\ell \parallel w_{ij}]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyRelu}(\vec{a}^\top [W X_i^\ell \parallel W X_k^\ell \parallel w_{ik}]))}, \quad (129)$$

with $\parallel$ 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

$$\exp(\text{LeakyRelu}(\vec{a}^\top [W X_i^\ell \parallel W X_j^\ell \parallel w_{ij}])) \rightarrow 0 \quad (130)$$

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

$$\exp(\text{LeakyRelu}(\vec{a}^\top [W X_i^\ell \parallel W X_j^\ell \parallel w_{ij}])) \rightarrow \infty \quad (131)$$

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

$$\alpha_{ij} \rightarrow 0, \quad (132)$$

since the denominator in (129) diverges. Hence in this case, propagation along $\mathcal{E}_{\text{reg.}}$ is essentially suppressed and features are effectively only propagated along $\mathcal{E}_{\text{high}}$.

E. Stability of latent representations generated by Laplace-transform based GNNs for graphs on the same node set

E.1. Node Level Stability

In this Section, we are concerned with two graphs G_1 and G_2 that share a common node set. We assume that their heat kernels are similar in the sense of

$$\|e^{-tL_1} - e^{-tL_2}\| \approx 0. \quad (133)$$

We are then interested in bounding the variation in latent representations generated by a GNN $\Phi_{\mathcal{W}, \mathcal{B}, \Psi}$ (as defined in Section 7) with weights $\mathcal{W}$, (potential) biases, and Laplace transform propagation matrices Ψ .

E.1.1. SPECTRAL LAPLACE-TRANSFORM METHODS

We first consider spectral Laplace Transform based spectral GNNs:

Theorem E.1. Let $\Xi_{\mathcal{W}, \mathcal{B}, \Psi}$ be a K -layer deep spectral graph convolutional architecture. Assume in each layer $1 \leq \ell \leq K$ that $\sum_i \|W_i^\ell\| \leq W$ and $\|B^\ell\| \leq B$ with weight- and bias-matrices $\{W^\ell, B^\ell\}$. Assume the non-linearity ρ is 1-Lipschitz. Choose a constant C such that $C \geq \|\Psi_i(L_1)\|, \|\Psi_i(L_2)\|$ for all Laplacian propagation matrices $\Psi_i(L_1), \Psi_i(L_2)$. W.l.o.g. assume $CW > 1$. With this, we have with $\delta = \max_{i \in I} \{\|\Psi_i(L_1) - \Psi_i(L_2)\|\}$ and input node-feature matrix X , that

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(L, X) - \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(\tilde{L}, X)\| \leq \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1} B \right) \right] \cdot \delta. \quad (134)$$

Proof. For simplicity in notation, let us denote the hidden representations in the network corresponding to L_2 by X_i^ℓ . With this, we note:

$$\|X_1^K - X_2^K\| \leq \sum_{i \in I} \|\psi_i(L_1) - \psi_i(L_2)\| \cdot \|X_1^{K-1}\| \cdot \|W_i^K\| + \sum_{i \in I} \|\psi_i(L_2)\| \cdot \|X_2^{K-1} - X_1^{K-1}\| \cdot \|W_i^K\| \quad (135)$$

$$\leq \delta W \|X_1^{K-1}\| + CW \|X_1^{K-1} - X_2^{K-1}\| \quad (136)$$

$$\leq \delta W \|X_1^{K-1}\| + CW \delta \|X_1^{K-2}\| + (CW)^2 \|X_2^{K-1} - X_1^{K-1}\| \quad (137)$$

$$\leq \frac{\delta}{C} \cdot \left(\sum_{\ell=1}^K (CW)^\ell \|X_1^{K-\ell}\| \right) \quad (138)$$

$$= \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \|X_1^j\| \right) \quad (139)$$

Hence we need to bound the quantity $\|X_1^j\|$ in terms of C, W, B and X .

We have

$$\|X_1^j\| \leq \sum_i \|\psi_i(L_1)\| \cdot \|X_1^{j-1}\| \cdot \|W_i^j\| + \|B^j\| \quad (140)$$

$$\leq CW \|X_1^{j-1}\| + B \quad (141)$$

$$\leq (CW)^2 \|X_1^{j-2}\| + CW B + B \quad (142)$$

$$\leq B \left(\sum_{k=0}^{j-1} (CW)^k \right) + (CW)^j \|X\| \quad (143)$$

$$= \begin{cases} B \frac{(CW)^j - 1}{CW - 1} + (CW)^j \|X\| & ; CW \neq 1 \\ jB + \|X\| & ; CW = 1 \end{cases}. \quad (144)$$

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

$$\|X_1^K - X_2^K\| \leq \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (jB + \|X\|) \right) \quad (145)$$

$$= \frac{\delta}{C} \cdot \left(K\|X\| + B \frac{K(K-1)}{2} \right). \quad (146)$$

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

$$\|X_1^K - X_2^K\| \leq \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \left[B \frac{(CW)^j - 1}{CW - 1} + (CW)^j \|X\| \right] \right) \quad (147)$$

$$(148)$$

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

$$\|X_1^K - X_2^K\| \leq \frac{\delta}{C} \cdot \left(\sum_{j=0}^{K-1} (CW)^{K-j} \left[B \frac{(CW)^j - 1}{CW - 1} + (CW)^j \|X\| \right] \right) \quad (149)$$

$$\leq \delta \cdot \frac{K(CW)^K}{C} \left[\frac{B}{CW - 1} + \|X\| \right]. \quad (150)$$

This proves the claim. $\square$

Next we note that we can estimate δ in terms of $\max_i \{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \}$.

Lemma E.1. *We have*

$$\delta \leq \max_i \left\{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \right\}. \quad (151)$$

Proof. We note that by definition (c.f. 8), we have

$$\psi_i(L) := \int_0^\infty e^{-tL} \hat{\psi}_i(t) dt. \quad (152)$$

From this we may infer

$$\|\psi_i(L_1) - \psi_i(L_2)\| = \left\| \int_0^\infty \hat{\psi}_i(t) [e^{-tL_1} - e^{-tL_2}] dt \right\| \leq \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt. \quad (153)$$

Taking the maximum over i indexing all propagation matrices, yields the claim. $\square$

Combining Lemma E.1 with Theorem E.1 then establishes a bound on node-wise latent embeddings in terms of the heat kernel distance $\|e^{-tL_1} - e^{-tL_2}\|$ for spectral Laplace transform based methods.

E.1.2. MESSAGE PASSING BASED LAPLACE TRANSFORM METHODS:

We next establish similar results for Message passing networks. To this end, we first introduce some notation. We continue to denote the node-feature matrix by X . As discussed in (10)&(11), we have as layer-wise update the implemented the scheme

$$m_{vu}^{(\ell)} = [\psi(L)]_{vu} \cdot \phi(h_v^{(k)}, h_u^{(k)}) \quad (154)$$

$$h_v^{(\ell+1)} = \sum_{u \in G} m_{vu}. \quad (155)$$

Denote the feature dimension of layer ℓ by d_ℓ . Hence we have $h_u^{(\ell)} \in \mathbb{R}^\ell$ and $\phi(h_v^{(\ell)}, h_u^{(\ell)}) \in \mathbb{R}^{d_{\ell+1}}$.

As shorthand notation, we collect the latent representations in layer (ℓ) into a feature matrix X^ℓ whose columns are given as

$$[X^\ell]_{u:} = h_u^{(\ell)}. \quad (156)$$

We then define the map $\Phi : \mathbb{R}^{N \times d_\ell} \rightarrow \mathbb{R}^{N \times N \times d_{\ell+1}}$ componentwise as

$$[\Phi(X^\ell)]_{vu:} = \phi([X^\ell]_{v:}, [X^\ell]_{u:}) \equiv \phi(h_v^\ell, h_u^\ell). \quad (157)$$

With this we find

$$[X^{\ell+1}]_{v:} = \sum_u [\psi(L)]_{vu} [\Phi(X^\ell)]_{vu:}. \quad (158)$$

This now allows us to prove:

Theorem E.2. Suppose we are given two graphs G_1, G_2 defined on a common node set. Denote the collection of node-wise latent representations generated by a K -layer deep Laplace transform based message passing network $\Xi_{\psi, \phi}$ by X_1^K, X_2^K respectively.

Suppose there is a constant C so that $C \geq \|\psi(L_1)\|, \|\psi(L_2)\|$. Assume that the message function ϕ is Lipschitz continuous:

$$\|\phi([X_1]_{v:}, [X_1]_{u:}) - \phi([X_2]_{v:}, [X_1]_{u:})\| \leq \frac{L}{4} (\|[X_1]_{v:}, -[X_2]_{v:}\| + \|[X_1]_{u:}, -[X_2]_{u:}\|). \quad (159)$$

Further assume that Φ is bounded, with $\|\Phi(\cdot)\| \leq D$. Then there is a constant C_Ξ so that

$$\|X_1^K - X_2^K\| \leq C_\Xi \cdot \|\psi(L_1) - \psi(L_2)\|. \quad (160)$$

Proof. We note:

$$[X_1^K]_{v:} - [X_2^K]_{v:} = \sum_u [\psi(L_1)]_{vu} [\Phi(X_1^{K-1})]_{vu:} - \sum_u [\psi(L_2)]_{vu} [\Phi(X_2^{K-1})]_{vu:} \quad (161)$$

$$= \left[\sum_u ([\psi(L_1)]_{vu} - \psi(L_2)_{vu}) [\Phi(X_1^{K-1})]_{vu:} \right] + \left[\sum_u [\psi(L_2)]_{vu} ([\Phi(X_1^{K-1})]_{vu:} - [\Phi(X_2^{K-1})]_{vu:}) \right]. \quad (162)$$

With this, we find

$$\|X_1^K - X_2^K\| \quad (163)$$

$$\leq \left(\sum_v \left\| \sum_u ([\psi(L_1)]_{vu} - \psi(L_2)_{vu}) [\Phi(X_1^{K-1})]_{vu:} \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (164)$$

$$+ \left(\sum_v \left\| \sum_u [\psi(L_2)]_{vu} ([\Phi(X_1^{K-1})]_{vu:} - [\Phi(X_2^{K-1})]_{vu:}) \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (165)$$

with the node weights (c.f. Section 2.1) denoted by $\{\mu_v\}_v$.

To bound the first term, we note:

$$\left(\sum_v \left\| \sum_u ([\psi(L_1)]_{vu} - \psi(L_2)_{vu}) [\Phi(X_1^{K-1})]_{vu} \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (166)$$

$$\leq D \cdot \left(\sum_{u,v} |[\psi(L_1)]_{vu} - \psi(L_2)_{vu}|^2 \mu_v \right)^{\frac{1}{2}} \quad (167)$$

$$\lesssim D \|\psi(L_1) - \psi(L_2)\| \cdot \left(\sum_v \mu_v \right)^{\frac{1}{2}} \quad (168)$$

$$= D \sqrt{\mu(G)} \|\psi(L_1) - \psi(L_2)\|, \quad (169)$$

Here we have bounded $|[\psi(L_1)]_{vu} - \psi(L_2)_{vu}| \lesssim \|\psi(L_1) - \psi(L_2)\|$.⁶

Let us next bound the second term. We have

$$\left(\sum_v \left\| \sum_u [\psi(L_2)]_{vu} ([\Phi(X_1^{K-1})]_{vu} - [\Phi(X_2^{K-1})]_{vu}) \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (170)$$

$$\leq C \cdot \left(\sum_{u,v} \|([\Phi(X_1^{K-1})]_{vu} - [\Phi(X_2^{K-1})]_{vu})\|^2 \mu_v \mu_u \right)^{\frac{1}{2}} \quad (171)$$

$$\leq CL \left(\sum_u \mu_u \right)^{\frac{1}{2}} \cdot \left(\sum_u \| [X_1^{K-1}]_{u:} - [X_2^{K-1}]_{u:} \|^2 \mu_u \right)^{\frac{1}{2}}. \quad (172)$$

$$= CL \sqrt{\mu(G)} \cdot \|X_1^{K-1} - X_2^{K-1}\|. \quad (173)$$

Putting the two terms together yields the iteration step

$$\|X_1^K - X_2^K\| \leq C_A \|\psi(L_1) - \psi(L_2)\| + C_B \|X_1^{K-1} - X_2^{K-1}\|, \quad (174)$$

with constants C_A, C_B as determined above. Iterating this through to $K = 0$ yields the following:

Let $d_K := \|X_1^K - X_2^K\|$ and $\Delta := \|\psi(L_1) - \psi(L_2)\|$. Assume the recursive bound

$$d_K \leq C_A \Delta + C_B d_{K-1}, \quad K \geq 1.$$

Furthermore we have the initial condition $d_0 = \|X_1^0 - X_2^0\| = \|X - X\| = 0$.

Iterating the inequality yields

$$\begin{aligned} d_K &\leq C_A \Delta + C_B d_{K-1} \\ &\leq C_A \Delta + C_B (C_A \Delta + C_B d_{K-2}) \\ &= C_A \Delta (1 + C_B) + C_B^2 d_{K-2} \\ &\vdots \\ &\leq C_A \Delta \sum_{j=0}^{K-1} C_B^j + C_B^K d_0. \end{aligned}$$

Using $d_0 = 0$, we obtain

$$\|X_1^K - X_2^K\| \leq C_A \|\psi(L_1) - \psi(L_2)\| \sum_{j=0}^{K-1} C_B^j.$$

⁶We are using the standard notation ' $\lesssim$ ' to denote smaller than, up to a fixed 'universal' constant.

Evaluating the geometric series gives the closed-form bound

$$\|X_1^K - X_2^K\| \leq \begin{cases} C_A \|\psi(L_1) - \psi(L_2)\| \frac{1 - C_B^K}{1 - C_B}, & C_B \neq 1, \\ K C_A \|\psi(L_1) - \psi(L_2)\|, & C_B = 1. \end{cases}$$

In particular, if $0 \leq C_B < 1$, then

$$\|X_1^K - X_2^K\| \leq \frac{C_A}{1 - C_B} \|\psi(L_1) - \psi(L_2)\|,$$

which provides a uniform bound independent of K . □

Combining the above Theorem with (153) then yields

Corollary E.3. *In the setting of Theorem E.2 we have*

$$\|X_1^K - X_2^K\| \leq C_\Xi \cdot \int_0^\infty \hat{\psi}(t) \|e^{-tL_1} - e^{-tL_2}\| dt = C_\Xi \max_i \left\{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \right\}. \quad (175)$$

Here the last equality follows since we are taking the maximum over all used propagation matrices, and only ever use a single one.

Hence we have established a bound on variations of node-wise latent embeddings in terms of the heat kernel distance $\|e^{-tL_1} - e^{-tL_2}\|$ for spectral Laplace transform based methods.

E.2. Graph Level Stability and Proof of Corollary 7.1

Next we promote the node-level stability results we derived in Subsection E.1 above to graph level stability results:

We may summarize Theorems E.1 & E.2 as: Given an input node-feature matrix X on graphs G_1, G_2 sharing a common node set, there exists constants C so that the output variation $\|X_1^K - X_2^K\|$ of K -layer deep Laplace transform based GNNs may be bounded as

$$\|X_1^K - X_2^K\| \leq C \max_i \left\{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \right\}. \quad (176)$$

We are now interested in how these stability results translate to graph level features. To this end we first specify our graph level aggregation method Ω :

Definition E.2. At the final (K^{th} -)layer We aggregate node-level embeddings $X^K \in \mathbb{R}^{N \times d_K}$ of individual nodes to graph-embeddings $\Omega(X) \in \mathbb{R}^{d_K}$ as $\Omega(X^K)_j = \sum_{i=1}^N |X_{ij}^K| \cdot \mu_i$. Here $\{\mu_i\}_i$ is the set of node-weights (c.f. Section 2.1).

Graph level features F_1, F_2 are then generated as $F_1 = \Omega(X_1^K)$ and $F_2 = \Omega(X_2^K)$.

With this we find:

Theorem E.4. *In the setting of (176), with graph level aggregation as in Definition E.2, we find*

$$\|F_1 - F_2\| \leq C \max_i \left\{ \int_0^\infty \hat{\psi}_i(t) \|e^{-tL_1} - e^{-tL_2}\| dt \right\}. \quad (177)$$

Proof. We only have to note that the aggregation method Ω is 1-Lipschitz (as a consequence of the reverse triangle inequality). Hence

$$\|F_1 - F_2\| = \|\Omega(X_1^K) - \Omega(X_2^K)\| \leq \|X_1^K - X_2^K\| \quad (178)$$

and the claim follows from 176. □

This then proves Corollary 7.1.

F. Stability of latent representations generated by Laplace-transform based GNNs for graphs on differing resolution scales

In this section, we establish bounds on differences in latent representations generated for Laplace-transform based GNNs when confronted with graphs describing the same underlying object at multiple resolutions. We briefly recall the setting:

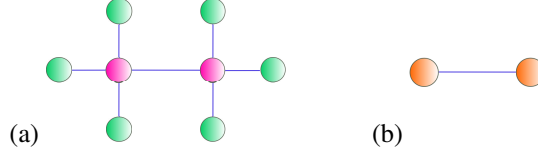


Figure 19. (a) Original graph G (b) Coarse grained $\underline{G}$

We have a high resolution graph G with associated Laplacian L and node-feature matrix X . We also have a lower resolution graph $\underline{G}$, with associated Laplacian $\underline{L}$ and node-feature matrix $\underline{X} := J^\downarrow X$ arising from the original node feature matrix X on G via projection to $\underline{G}$. Here we have made use of the projection operator $J^\downarrow$ introduced in Section 6.1 which averages node information over clusters that are condensed into supernodes. Interpolation $J^\uparrow$ instead assigns information at each supernode in $\underline{G}$ to every node making up the corresponding cluster in G .

We have (Koke, 2026b) (c.f. also Appendix H) that $J^\downarrow J^\uparrow = I_{\underline{G}}$ (i.e. equal to the identity matrix on $\underline{G}$).

F.1. Node Level Results

We are then initially interested in the following question: Suppose we have a node feature matrix X on the graph G . We can generate node-level latent embeddings $\Xi(X)$ by feeding this node-wise information into a (Laplace-transform based) node-level GNN Ξ . How different is this outcome from projecting X onto the coarser graph $\underline{G}$, running the GNN there to generate node-level embeddings $\Xi(J^\downarrow X)$ and then interpolating the generated embeddings back up to G via the interpolation operator $J^\uparrow$. That is to say, we are interested in the difference $\|\Xi(X) - J^\uparrow \Xi(J^\downarrow X)\|$.

F.1.1. SPECTRAL LAPLACE-TRANSFORM METHODS

We begin by establishing $\|\Xi(X) - J^\uparrow \Xi(J^\downarrow X)\| \leq C \cdot \max_i \left\{ \int_0^\infty |\hat{\psi}_i(t)| \cdot \|e^{-tL} - J^\uparrow e^{-t\underline{L}} J^\downarrow\| \right\}$ for spectral methods:

Theorem F.1. Let $\Xi_{\mathcal{W}, \mathcal{B}, \Psi}$ be a K -layer deep Laplace transform based network. Assume that $\sum_{i \in I} \|W_i^\ell\| \leq W$ and $\|B^k\| \leq B$. Choose $C \geq \|\Psi_i(L)\|, \|\Psi_i(\underline{L})\|$ ($i \in I$) and w.l.o.g. assume $CW > 1$. Set $\max_{i \in I} \{\|\psi_i(L) - J^\uparrow \psi_i(\underline{L}) J^\downarrow\|\} = \delta$. With this, we have that

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(X) - J^\uparrow \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(J^\downarrow X)\| \leq \left[K \cdot C^K W^{K-1} \cdot \left(\|X\| + \frac{1}{CW-1} B \right) \right] \cdot \delta. \quad (179)$$

Proof. Let us define

$$\underline{X} := J^\downarrow X. \quad (180)$$

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

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

$$X^\ell = \rho \left(\sum_{i \in I} \psi_i X^{\ell-1} W_i^\ell + B^\ell \right) \quad (181)$$

and

$$\underline{X}^\ell = \rho \left(\sum_{i \in I} \underline{\psi}_i \underline{X}^{\ell-1} W_i^\ell + \underline{B}^\ell \right). \quad (182)$$

Since bias terms are proportional to constant vectors on the graphs, we have

$$J^\downarrow B = \underline{B} \quad (183)$$

and

$$J^\uparrow \underline{B} = B \quad (184)$$

for bias matrices B and $\underline{B}$ in networks deployed on G and $\underline{G}$ respectively.

We then have

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(X) - J^\uparrow \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(J^\downarrow X)\| \quad (185)$$

$$= \|X^K - J^\uparrow \underline{X}^K\| \quad (186)$$

$$= \left\| \rho \left(\sum_{i \in I} \psi_i X^{K-1} W_i^K + B^K \right) - J^\uparrow \rho \left(\sum_{i \in I} \psi_i \underline{X}^{K-1} W_i^K + \underline{B}^L \right) \right\| \quad (187)$$

$$= \left\| \rho \left(\sum_{i \in I} \psi_i X^{K-1} W_i^K + B^K \right) - \rho \left(J^\uparrow \sum_{i \in I} \tilde{\psi}_i \underline{X}^{K-1} W_i^K + B^L \right) \right\| \quad (188)$$

$$(189)$$

Here we used the fact that ρ and $J^\uparrow$ commute (since $J^\uparrow$ is an assignment matrix and has all non-zero entries equal to one). Using the fact that the non-linearity $\rho(\cdot)$ is 1-Lipschitz-continuous we can establish

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(X) - J^\uparrow \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(J^\downarrow X)\| \quad (190)$$

$$\leq \left\| \rho \left(\sum_{i \in I} \psi_i X^{K-1} W_i^K + B^K \right) - \rho \left(J^\uparrow \sum_{i \in I} \psi_i \underline{X}^{K-1} W_i^K + B^K \right) \right\| \quad (191)$$

$$\leq \left\| \sum_{i \in I} \psi_i X^{K-1} W_i^K + B^K - J^\uparrow \sum_{i \in I} \psi_i \underline{X}^{K-1} W_i^K + B^K \right\|. \quad (192)$$

Since $J^\downarrow J^\uparrow = Id_{\underline{G}}$, we have

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(X) - J^\uparrow \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(J^\downarrow X)\| \quad (193)$$

$$\leq \left\| \sum_{i \in I} \psi_i X^{K-1} W_i^K - \sum_{i \in I} (J^\uparrow \psi_i J^\downarrow) J^\uparrow \underline{X}^{K-1} W_i^K \right\| \quad (194)$$

From this, we find (using $\|J^\uparrow\|, \|J^\downarrow\| \leq 1$), that

$$\|\Xi_{\mathcal{W}, \mathcal{B}, \Psi}(X) - J^\uparrow \Xi_{\mathcal{W}, \mathcal{B}, \Psi}(J^\downarrow X)\| \quad (195)$$

$$\leq \left\| \sum_{i \in I} \psi_i X^{K-1} W_i^K - \sum_{i \in I} (J^\uparrow \psi_i J^\downarrow) J^\uparrow \underline{X}^{K-1} W_i^K \right\| \quad (196)$$

$$\leq \left\| \sum_{i \in I} (\psi_i - J^\uparrow \psi_i J^\downarrow) X^{K-1} W_i^K \right\| + \sum_{i \in I} \|J^\uparrow \psi_i J^\downarrow\| \cdot \|J^\uparrow \underline{X}^{K-1} - X^{K-1}\| \cdot \|W_i^K\| \quad (197)$$

$$\leq \left\| \sum_{i \in I} (\psi_i - J^\uparrow \psi_i J^\downarrow) X^{K-1} W_i^K \right\| + CW \cdot \|J^\uparrow \underline{X}^{K-1} - X^{K-1}\| \quad (198)$$

$$\leq \sum_{i \in I} \|(\psi_i - J^\uparrow \psi_i J^\downarrow)\| \cdot \|X^{K-1}\| \cdot \|W_i^K\| + CW \cdot \|\tilde{J}^\downarrow \underline{X}^{K-1} - X^{K-1}\| \quad (199)$$

$$\leq \delta \cdot \|X^{K-1}\| W + CW \cdot \|\tilde{J}^\uparrow \underline{X}^{K-1} - X^{K-1}\| \quad (200)$$

$$(201)$$

Arguing as in the proof of Theorem E.1 then yields the claim.

□

F.1.2. MESSAGE PASSING BASED LAPLACE TRANSFORM METHODS:

We next establish similar results for message passing networks. Sticking with the notation of the preceding Subsection F.1.1, we are hence interested in bounding $X^K - J^\dagger \underline{X}^K$, with $\underline{X}$ being the node feature matrix generated after a K -layer Laplace transform based MPNN.

Theorem F.1. *Assume there is a constant C so that $C \geq \|\psi(L)\|, \|\psi(\underline{L})\|$. Assume that the message function ϕ is Lipschitz continuous:*

$$\|\phi([X_1]_{v:}, [X_1]_{u:}) - \phi([X_2]_{v:}, [X_1]_{u:})\| \leq \frac{L}{4} (\|[X_1]_{v:}, -[X_2]_{v:}\| + \|[X_1]_{u:}, -[X_2]_{u:}\|). \quad (202)$$

Further assume that Φ is bounded, with $\|\Phi(\cdot)\| \leq D$. Then there is a constant C_Ξ so that

$$\|X^K - J^\dagger \underline{X}^K\| \leq C_\Xi \cdot \int_0^\infty \hat{\psi}(t) \|e^{-tL} - J^\dagger e^{-t\underline{L}} J^\dagger\|. \quad (203)$$

Proof. We note:

$$X_{v:}^K - [J^\dagger \underline{X}^K]_{v:} = X_{v:}^K - \sum_{a \in \underline{G}} J_{va}^\dagger X_{a:}^K = \sum_{u \in G} [\psi(L)]_{vu} [\Phi(X^{K-1})]_{vu:} - \sum_{a, b \in \underline{G}} J_{va}^\dagger [\psi(\underline{L})]_{ab} [\Phi(\underline{X}^{K-1})]_{ab:}. \quad (204)$$

Creatively adding zero yields

$$X_{v:}^K - [J^\dagger \underline{X}^K]_{v:} \quad (205)$$

$$= \sum_{u \in G} [\psi(L)]_{vu} [\Phi(X^{K-1})]_{vu:} - \sum_{u \in G} [\psi(L)]_{vu} [\Phi(J^\dagger \underline{X}^{K-1})]_{vu:} \quad (206)$$

$$+ \sum_{u \in G} [\psi(L)]_{vu} [\Phi(J^\dagger \underline{X}^{K-1})]_{vu:} - \sum_{a, b \in \underline{G}} J_{va}^\dagger [\psi(\underline{L})]_{ab} [\Phi(\underline{X}^{K-1})]_{ab:}. \quad (207)$$

For the first term, we note

$$\left(\sum_{v \in G} \left\| \sum_{u \in G} [\psi(L)]_{vu} [\Phi(X^{K-1})]_{vu:} - \sum_{u \in G} [\psi(L)]_{vu} [\Phi(J^\dagger \underline{X}^{K-1})]_{vu:} \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (208)$$

$$\leq C \cdot \left(\sum_{v, u \in G} \sum_{u \in G} \left\| [\Phi(X^{K-1})]_{vu:} - [\Phi(J^\dagger \underline{X}^{K-1})]_{vu:} \right\|^2 \mu_v \right)^{\frac{1}{2}} \quad (209)$$

$$\leq CL \left(\sum_u \mu_u \right)^{\frac{1}{2}} \cdot \left(\sum_u \left\| [X^{K-1}]_{u:}, -[J^\dagger \underline{X}]_{u:} \right\|^2 \mu_u \right)^{\frac{1}{2}}. \quad (210)$$

$$= CL \sqrt{\mu(G)} \cdot \|X^{K-1} - J^\dagger \underline{X}^{K-1}\|. \quad (211)$$

Hence let us consider the second term. By Lemma F.2 below, we then have

$$\sum_{v \in G} \mu_g \left\| \sum_{u \in G} [\psi(L)]_{vu} [\Phi(J^\uparrow \underline{X}^{K-1})]_{vu} - \sum_{a, b \in \underline{G}} J_{va}^\uparrow [\psi(\underline{L})]_{ab} [\Phi(\underline{X}^{K-1})]_{ab} \right\|^2 \quad (212)$$

$$= \sum_{v \in G} \mu_g \left\| \sum_{u \in G} [\psi(L)]_{vu} \sum_{a, b \in \underline{G}} J_{va}^\uparrow J_{ub}^\uparrow [\Phi(\underline{X}^{K-1})]_{ab} - \sum_{a, b \in \underline{G}} J_{va}^\uparrow [\psi(\underline{L})]_{ab} [\Phi(\underline{X}^{K-1})]_{ab} \right\|^2 \quad (213)$$

$$= \sum_{v \in G} \mu_g \left\| \sum_{a, b \in \underline{G}} J_{va}^\uparrow [\psi(L) J^\uparrow]_{vb} [\Phi(\underline{X}^{K-1})]_{ab} - \sum_{a, b \in \underline{G}} J_{va}^\uparrow [\psi(\underline{L})]_{ab} [\Phi(\underline{X}^{K-1})]_{ab} \right\|^2 \quad (214)$$

$$= \sum_{v \in G} \mu_g \left\| \sum_{a, b \in \underline{G}} J_{va}^\uparrow [\psi(L) J^\uparrow]_{vb} [\Phi(\underline{X}^{K-1})]_{ab} - \sum_{a, b \in \underline{G}} J_{va}^\uparrow [J^\uparrow \psi(\underline{L})]_{vb} [\Phi(\underline{X}^{K-1})]_{ab} \right\|^2 \quad (215)$$

$$\leq D^2 \sum_{v \in G} \mu_g \left(\sum_{a, b \in \underline{G}} J_{va}^\uparrow |[\psi(L) J^\uparrow]_{vb} - [J^\uparrow \psi(\underline{L})]_{vb}| \right)^2 \quad (216)$$

$$\lesssim D^2 \sum_{v \in G} \mu_g \left(\sum_{a \in \underline{G}} J_{va}^\uparrow \|\psi(L) J^\uparrow - J^\uparrow \psi(\underline{L})\| \right)^2 \quad (217)$$

$$\leq D^2 \mu(G) \|\psi(L) J^\uparrow - J^\uparrow \psi(\underline{L})\|^2 \quad (218)$$

$$\lesssim D^2 \mu(G) \|\psi(L) - J^\uparrow \psi(\underline{L}) J^\downarrow\|^2. \quad (219)$$

Here the third step follows from the fact that $J_{va}^\uparrow [J^\uparrow \psi(\underline{L})]_{vb} = J_{va}^\uparrow [\psi(\underline{L})]_{ab}$. The last step follows from the fact that $J^\downarrow$ is surjective.

Recursively, we thus have established that

$$\|X^K - J^\uparrow \underline{X}^K\| \leq C_A \|\psi(L) - J^\uparrow \psi(\underline{L}) J^\downarrow\| + C_B \|X^{K-1} - J^\uparrow \underline{X}^{K-1}\|. \quad (220)$$

We may unroll the recursion to obtain

$$\|X^K - J^\uparrow \underline{X}^K\| \leq C_A \sum_{i=0}^{K-1} C_B^i \|\psi(L) - J^\uparrow \psi(\underline{L}) J^\downarrow\| + C_B^K \|X^0 - J^\uparrow \underline{X}^0\|. \quad (221)$$

If $C_B \neq 1$, the geometric sum yields

$$\|X^K - J^\uparrow \underline{X}^K\| \leq C_A \frac{1 - C_B^K}{1 - C_B} \|\psi(L) - J^\uparrow \psi(\underline{L}) J^\downarrow\| + C_B^K \|X^0 - J^\uparrow \underline{X}^0\|. \quad (222)$$

Finally we recall from Section 7.1, that the initial input X^0 into the Laplace-transform message passing layers arises from the input features X via an initial Laplace-transform propagation step as

$$X^0 = \psi(L)X. \quad (223)$$

Thus we also find

$$J^\uparrow \underline{X}^0 = J^\uparrow \psi(\underline{L}) [J^\downarrow X]. \quad (224)$$

From this we may infer that

$$\|X^0 - J^\uparrow \underline{X}^0\| = \|\psi(L)X - J^\uparrow \psi(\underline{L}) J^\downarrow X\| \leq \|\psi(L) - J^\uparrow \psi(\underline{L})\| \cdot \|X\|. \quad (225)$$

Hence both summands in (222) may be bounded in terms of $\|\psi(L) - J^\uparrow \psi(\underline{L})\|$. Using

$$\|\psi(L) - J^\uparrow \psi(\underline{L})\| \leq \int_0^\infty \|\hat{\psi}(t)\| e^{-tL} - J^\uparrow e^{-t\underline{L}} J^\downarrow \|dt, \quad (226)$$

we have thus established the claim. $\square$

Finally we provide the Lemma used in the proof above:

Lemma F.2. *We have*

$$[\Phi(J^\uparrow \underline{X}^\ell)]_{vu} = \sum_{a,b \in \underline{G}} J_{va}^\uparrow J_{ub}^\uparrow [\Phi(\underline{X}^\ell)]_{ab}. \quad (227)$$

Proof. Indeed, this follows because entries of $J^\uparrow$ satisfy $J_{ua}^\uparrow \in \{0, 1\}$ and every node in G gets assigned information exactly from only one node in $\underline{G}$. Thus the sum $\sum_{a,b \in \underline{G}} J_{va}^\uparrow J_{ub}^\uparrow [[\Phi(\underline{X}^\ell)]_{ab}]$ contains precisely *one* non-zero summand. This summand in turn, corresponds to those indices $a, b \in \underline{G}$ which are mapped to v and u respectively under $J^\uparrow$. Since entries in $J^\uparrow$ are binary (and hence do not diminish or blow up signal if they are non-zero) the claim follows. $\square$

F.2. Graph Level Results and Proof of Theorem 7.1

It remains to translate the results of Theorems F.1 & F.1 to the graph level. To this end, we first note the following:

Lemma F.2. *With the graph level aggregation function Ω as in Definition E.2, we have*

$$\Omega(\underline{X}) = \Omega(J^\uparrow \underline{X}). \quad (228)$$

Proof. This fact follows immediately from the fact that $J^\uparrow$ assigns the information present at the supernode $a \in \underline{G}$ to all nodes u in the cluster within G that correspond to the supernode a , together with the fact that the node-weight μ_a of the node $a \in \underline{G}$ is the sum over all of the node weights μ_u of nodes u in the cluster within G that corresponds to the supernode a under coarse graining (c.f. also (290) in Appendix H). $\square$

Using the above lemma, we find

$$\|\underline{F} - F\| = \|\Omega(\underline{X}^K) - \Omega(X^K)\| = \|\Omega(J^\uparrow \underline{X}^K) - \Omega(X^K)\|. \quad (229)$$

Using the 1-Lipschitzness of the aggregation function Ω then yields

$$\|\Omega(J^\uparrow \underline{X}^K) - \Omega(X^K)\| \leq \|J^\uparrow \underline{X}^K - X^K\|. \quad (230)$$

Together with Theorems F.1 & F.1 we have thus proved:

Theorem F.3. *Let F and $\underline{F}$ be the latent embeddings generated for a graph G its coarsified version $\underline{G}$ by a (spectral or message passing) network employing Laplace transform propagation, as outlined in Section 7.1. With $\{\Psi_i(L) = \int_0^\infty \hat{\psi}_i(t) e^{-tL} dt\}_{i \in I}$ the collection of all Laplace-transform propagation matrices used within the network, we have $\|F - \underline{F}\| \leq C \cdot \max_i \left\{ \int_0^\infty |\hat{\psi}_i(t)| \cdot \|e^{-tL_\omega} - J^\uparrow e^{-t\underline{L}} J^\downarrow\| \right\}$.*

G. Additional Experimental Considerations

G.1. Additional details on initial Coarse Graining Experiment

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^{Cmb} , whose off-diagonal elements

$$C_{ij}^{\text{Cmb}} = \frac{Z_i Z_j}{|R_i - R_j|} \quad (231)$$

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

$$C_{ij}^{Cmb} = \frac{Z_i Z_j}{|R_i - R_j|} \quad (232)$$

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 (232).

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 (232) 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

$$X = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 0 \cdots \\ 1 & 0 & \cdots & 0 & 0 & 0 \cdots \\ 1 & 0 & \cdots & 0 & 0 & 0 \cdots \\ 1 & 0 & \cdots & 0 & 0 & 0 \cdots \\ 1 & 0 & \cdots & 0 & 0 & 0 \cdots \end{pmatrix} \quad (233)$$

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

$$\mu = 6 + 1 + 1 + 1 + 1 = 10. \quad (234)$$

The feature matrix

$$\underline{X} = J^\downarrow X \quad (235)$$

is a single row-vector given as

$$\underline{X} = \left(\frac{4}{10}, 0, \cdots, 0, \frac{6}{10}, 0, \cdots \right). \quad (236)$$

Experimental Setup: We randomly select 1500 molecules for testing and train on the remaining graphs. On QM7 we run experiments for 5 different 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), GATv2 (Brody et al., 2022)) GIN ((Xu et al., 2019) (together with its edge-weight enabled version (Hu et al., 2020) where applicable)) and **multi-scale** architectures (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 individual 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 ChebNet, we set the polynomial order to $K = 2$ as is customary. Lanczos 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.

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

For the $\text{Spectral}_{\text{Exp.}}$ -, we set $t = 1$ and build filters using the $k = 1$ and $k = 2$ atoms in $\Psi^{\text{Exp}} = \{e^{-(kt_0)z}\}_{k \in \mathbb{N}}$. For $\text{MPNN}_{\text{Res.}}$ we use the $k = 1$ atom as ψ .

As aggregation, we employ the graph level feature aggregation scheme discussed in Definition E.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.

G.2. Further discussions on stability results of Table 3

We can further understand the small difference of latent embeddings using Theorem 7.1. Let us use the shorthand notation

$$\eta(t) = \|e^{-Lt} - J^\dagger e^{-tL} J^\downarrow\|. \quad (237)$$

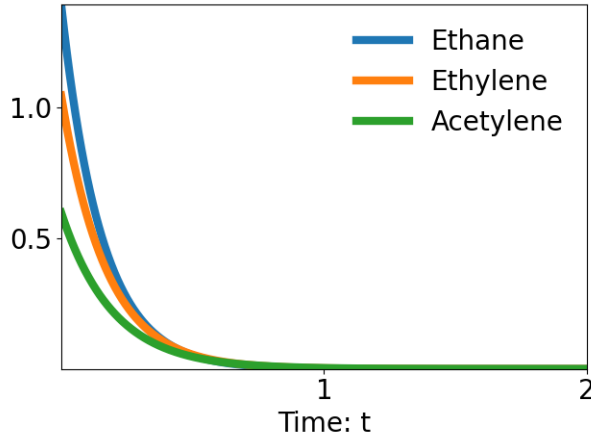


Figure 20. $\|e^{-Lt} - J^\dagger e^{-tL} J^\downarrow\|$ for molecules in QM7

Exemplarily considering exponential propagation matrices (cf. Section 6) we have (with $t_k = k$) that $\int_0^\infty |\hat{\psi}_k(t)| \|e^{-tL} - J^\dagger e^{-tL} J^\downarrow\| dt = \|e^{-kL} - J^\dagger e^{-kL} J^\downarrow\|$, we thus have $\|F - \underline{F}\| \lesssim \max_{k \geq 1} |\eta(k)|$. Investigating the differences $\eta(t) = \|e^{-tL} - J^\dagger e^{-tL} J^\downarrow\|$ of diffusion flows, we find that $\eta(t)$ drops to zero fast, as exemplarily plotted in Fig. 20 for the first few molecules of QM7. In particular $|\eta(k)|_{k \geq 1} \approx 0$.

Using this as an upper bound in Theorem 7.1 shows that embeddings $F, \underline{F}$ of graphs describing the same molecule at different resolution scales are similar. This explains the ability to generalize between scales.

G.3. 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. 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.

Experiment: Varying the Connectivity within the Communities: We fix the parameters c_{number} , c_{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 21 below, we then plot the difference of feature vectors generated by models 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.

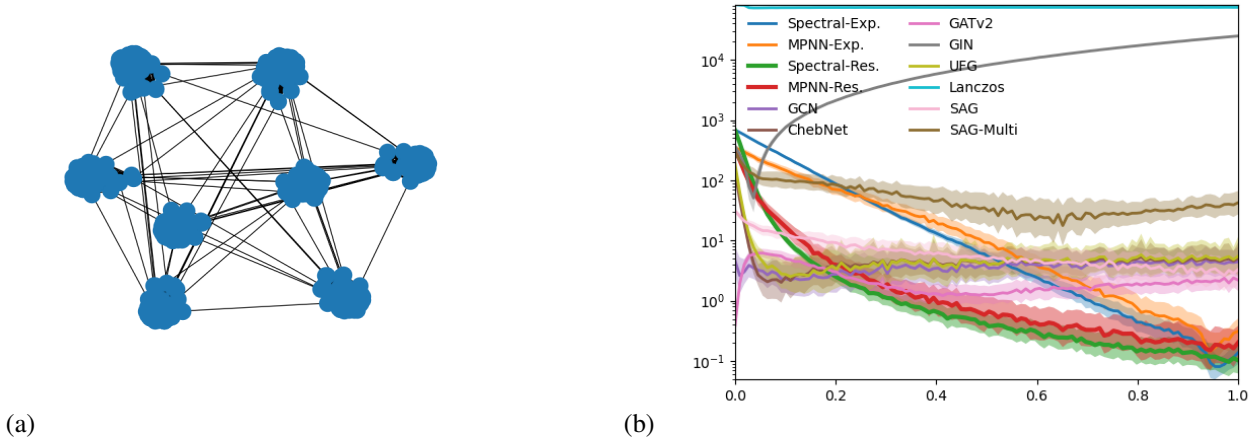


Figure 21. (a) Example Graph (b) Varying the parameter $p_{\text{connect}} \in [0, 1]$ for fixed $c_{\text{size}} = 60$, $p_{\text{inter}} = 2/c_{\text{size}}^2$ and $c_{\text{number}} = 12$.

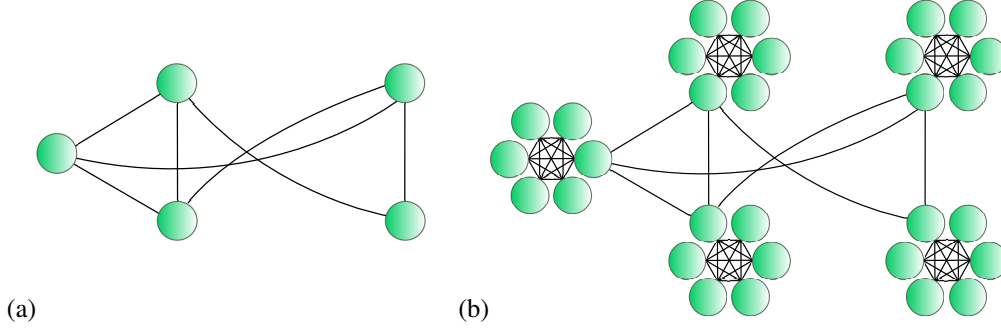
We have chosen $p_{\text{inter}} = 2/c_{\text{size}}^2$ so that – on average – *clusters* are connected by two edges. The choice of two edges (as opposed to 1, 3, 4, 5, ...) between clusters is not important.

G.4. Node Level Transferability and Graphs with Varying Connectivity

Here, 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.

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., 2019) on which our codebase for this experiment 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 $\text{lr} = 0.1$. 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


 Figure 22. Individual nodes (a) replaced by k -cliques (b)

over $\lambda \in \{0.0001, 0.0005\}$. We choose a two-layer deep convolutional architecture with the dimensions of hidden features optimized over

$$K_\ell \in \{32, 64, 128\}. \quad (238)$$

In addition to the hyperparameters specified above, some baselines have additional hyperparameters, which we detail here: ChebNet 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 (238) as for baselines. We empirically observed in the setting of *unweighted* graphs, that rescaling the Laplacian as $L_{nf} := \frac{1}{c_{nf}}L$ with a normalizing factor c_{nf} on which we base LTF 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 $\lambda \in \{0.14, 0.15, 0.2, 0.25\}$.

G.5. The Setting of Graphs discretizing a common ambient Manifold

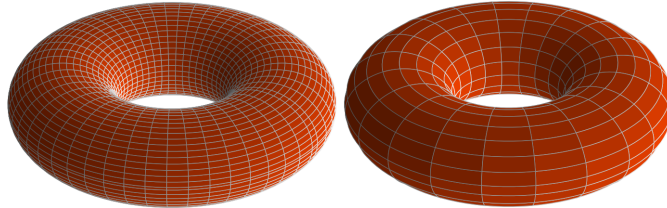


Figure 23. Distinct Torus Discretizations

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 a prime setting for studying transferability. Counter to previous works (Levie et al., 2019a; Wang et al., 2021), our framework here allows to derive transferability guarantees beyond the settings of bandlimited signals and probabilistic guarantees.

We consider the setting of two graphs G_1, G_2 discretely approximating the same manifold (c.f. e.g. Fig. 23). 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 $\|e^{-t\Delta_{\mathcal{M}}} - J_i^\uparrow e^{-tL_i} J_i^\downarrow\| \leq \delta_i$ in heat 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 below, we have in this setting:

$$\|e^{-tL_1} - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \lesssim (\delta_1 + \delta_2) \quad (239)$$

Together with Theorem 7.1, this then also implies

$$\|F_1 - F_2\| \lesssim (\delta_1 + \delta_2) \quad (240)$$

for the generated latent embeddings of the two graph approximations G_1 and G_2 . Below we show using the explicit example of the torus, that as the approximating mesh becomes finer and finer (i.e. the number of nodes N increases), the corresponding approximation error δ tends to zero. Equation (240) then guarantees that the corresponding graph level latent embeddings form a Cauchy sequence. Since the latent space $\mathbb{R}^d$ is complete, we thus have convergence of the latent embeddings F_N (indexed by the mesh size N) towards a limit embedding ($F_N \rightarrow F_\infty$), with the limit embedding F_∞ corresponding to the latent embedding our Laplace-transform based model would generate if we could to deploy it on the continuous manifold $\mathcal{M}$.

Hence let us further discuss the setting of two graphs G_i discretizing the same ambient manifold $\mathcal{M}$ in the sense of

$$\|J_i^\uparrow e^{-tL_i} J_i^\downarrow - e^{-t\Delta_{\mathcal{M}}}\| \leq \delta. \quad (241)$$

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

$$\|e^{-tL_1} - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \quad (242)$$

$$= \|e^{-tL_1} - J_1^\downarrow e^{-t\Delta_{\mathcal{M}}} J_1^\uparrow + J_1^\downarrow (\Delta_{\mathcal{M}} + Id)^{-1} J_1^\uparrow - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \quad (243)$$

$$\leq \|e^{-tL_1} - J_1^\downarrow e^{-t\Delta_{\mathcal{M}}} J_1^\uparrow\| + \|J_1^\downarrow e^{-t\Delta_{\mathcal{M}}} J_1^\uparrow - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \quad (244)$$

$$(245)$$

We note

$$\|e^{-tL_1} - J_1^\downarrow e^{-t\Delta_{\mathcal{M}}} J_1^\uparrow\| \quad (246)$$

$$= \|J_1^\downarrow J_1^\uparrow e^{-tL_1} J_1^\downarrow J_1^\uparrow - J_1^\downarrow e^{-t\Delta_{\mathcal{M}}} J_1^\uparrow\| \quad (247)$$

$$\leq \|J_1^\downarrow\| \|J_1^\uparrow\| \cdot \|e^{-tL_1} - J_1^\uparrow e^{-t\Delta_{\mathcal{M}}} J_1^\downarrow\| \lesssim \delta. \quad (248)$$

We consider:

$$\|e^{-t\Delta_{\mathcal{M}}} - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \quad (249)$$

$$\leq \|J_1^\downarrow\| \|J_1^\uparrow\| \cdot \|e^{-t\Delta_{\mathcal{M}}} - J_2^\uparrow e^{-tL_2} J_2^\downarrow\| \quad (250)$$

$$\lesssim \|e^{-t\Delta_{\mathcal{M}}} - J_2^\uparrow e^{-tL_2} J_2^\downarrow\| \leq \delta. \quad (251)$$

$$(252)$$

Hence we have indeed established

$$\|e^{-tL_1} - (J_1^\downarrow J_2^\uparrow) e^{-tL_2} (J_2^\downarrow J_1^\uparrow)\| \lesssim 2\delta. \quad (253)$$

Next let us consider an explicit example.

To this end, let us revisit the torus-setting introduced in Fig. 12. To see the origin of the estimate (239) in the torus setting, we begin by recalling that the standard torus $\mathbb{T}$ arises as the cartesian product of two circles S^1 of circumference 2π :

$$\mathbb{T} = S^1 \times S^1. \quad (254)$$

Let us parametrize these circles via angles $0 \leq \theta_1, \theta_2 \leq 2\pi$. The Laplacian on $\mathbb{T}$ can then be written as

$$\Delta_{\mathbb{T}} = -\partial_{\theta_1}^2 - \partial_{\theta_2}^2. \quad (255)$$

A set of corresponding normalized eigenfunctions are given as

$$\phi_{k_1, k_2} = \frac{1}{2\pi} e^{-ik_1\theta_1} e^{-ik_2\theta_2} \quad (256)$$

with corresponding eigenvalues

$$\lambda_{k_1, k_2} = k_1^2 + k_2^2 \quad (257)$$

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 $0 \leq k \leq N$. 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 $\mathbb{T}_N$ of $\mathbb{T}$ 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 L_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

$$(L_N x)_{ab} = \frac{N^2}{(2\pi)^2} (4x_{ab} - x_{(a+1)b} - x_{(a-1)b} - x_{a(b+1)} - x_{a(b-1)}). \quad (258)$$

Henceforth we will adopt the notation $x(a, b) \equiv x_{ab}$.

Normalized eigenvectors for this Laplacian L_N on $\mathbb{T}_N$ are given as

$$\phi_{k_1, k_2}^N = \frac{1}{2\pi} e^{-i\frac{2\pi k_1}{N}a} e^{-i\frac{2\pi k_2}{N}b} \quad (259)$$

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

$$\lambda_{k_1, k_2}^N = \frac{N^2}{\pi^2} \left[\sin^2 \left(\frac{\pi}{N} \cdot k_1 \right) + \sin^2 \left(\frac{\pi}{N} \cdot k_2 \right) \right]. \quad (260)$$

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 $\bar{f}$ defined on $\mathbb{T}$ by defining

$$\bar{f}(\theta_1, \theta_2) = f(a, b) \quad (261)$$

whenever $\frac{2\pi}{N}(a-1) \leq \theta_1 \leq \frac{2\pi}{N}a$ and $\frac{2\pi}{N}(b-1) \leq \theta_2 \leq \frac{2\pi}{N}b$.

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 $t > 0$)

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-tL_N} J^\downarrow\| \rightarrow 0 \quad (262)$$

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

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-tL_N} J^\downarrow\| = \left\| \sum_{k_1, k_2 \in \mathbb{Z}} e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} - \sum_{-\frac{N-1}{2} \leq p_1, p_2 \leq \frac{N-1}{2}} e^{-\lambda_{k_1, k_2} t} P_{p_1, p_2}^N \right\|. \quad (263)$$

From this we observe

$$\|e^{-t\Delta_{\mathbb{T}}} - J^\uparrow e^{-tL_N} J^\downarrow\| = \left\| \sum_{k_1, k_2 \in \mathbb{Z}} e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} - \sum_{-\frac{N-1}{2} \leq p_1, p_2 \leq \frac{N-1}{2}} e^{-\lambda_{p_1, p_2}^N t} P_{p_1, p_2}^N \right\| \quad (264)$$

$$\leq \left\| \sum_{\frac{N-1}{2} < |k_1|, |k_2|} e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} \right\| + \left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} \left(e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} - e^{-\lambda_{k_1, k_2}^N t} P_{k_1, k_2}^N \right) \right\| \quad (265)$$

For the first summand, we already have

$$\left\| \sum_{\substack{\frac{N-1}{2} < |k_1|, |k_2|}} e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} \right\| \leq e^{-t \frac{(N-1)^2}{2}}. \quad (266)$$

Hence let us investigate the second summand. We note

$$\left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} \left(e^{-\lambda_{k_1, k_2} t} P_{k_1, k_2} - e^{-\lambda_{k_1, k_2}^N t} P_{k_1, k_2}^N \right) \right\| \quad (267)$$

$$\leq \left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} \left(e^{-\lambda_{k_1, k_2} t} - e^{-\lambda_{k_1, k_2}^N t} \right) P_{k_1, k_2}^N \right\| + \left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} e^{-\lambda_{k_1, k_2} t} (P_{k_1, k_2} - P_{k_1, k_2}^N) \right\| \quad (268)$$

For the first summand we note

$$\left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} \left(e^{-\lambda_{k_1, k_2} t} - e^{-\lambda_{k_1, k_2}^N t} \right) P_{k_1, k_2}^N \right\| \quad (269)$$

$$= \sup_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} \left| e^{-\lambda_{k_1, k_2} t} - e^{-\lambda_{k_1, k_2}^N t} \right| \quad (270)$$

$$= \sup_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} e^{-t(k_1^2 + k_2^2)} \left| 1 - e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_1 \right) - k_1^2 \right)} e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_2 \right) - k_2^2 \right)} \right| \quad (271)$$

We note

$$\left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k \right) - k^2 \right) = \mathcal{O} \left(\frac{k^4}{N^2} \right). \quad (272)$$

Using

$$\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} N^{\frac{1}{3}} \right) \lesssim N^{\frac{2}{3}} \quad (273)$$

we note

$$\sup_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} e^{-t(k_1^2 + k_2^2)} \left| 1 - e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_1 \right) - k_1^2 \right)} e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_2 \right) - k_2^2 \right)} \right| \quad (274)$$

$$\leq \sup_{|k_1|, |k_2| \leq N^{\frac{1}{3}}} e^{-t(k_1^2 + k_2^2)} \left| 1 - e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_1 \right) - k_1^2 \right)} e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_2 \right) - k_2^2 \right)} \right| \quad (275)$$

$$+ \sup_{|k_1|, |k_2| > N^{\frac{1}{3}}} e^{-t(k_1^2 + k_2^2)} \left| 1 - e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_1 \right) - k_1^2 \right)} e^{-t \left(\frac{N^2}{\pi^2} \sin^2 \left(\frac{\pi}{N} k_2 \right) - k_2^2 \right)} \right| \quad (276)$$

$$\leq e^{-t(2N^{\frac{2}{3}})} + e^{-t(2N^{\frac{2}{3}})} + e^{-t(N^{\frac{2}{3}})}. \quad (277)$$

Hence it remains to bound the second summand in (267). We note

$$\left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} e^{-\lambda_{k_1, k_2} t} (P_{k_1, k_2} - P_{k_1, k_2}^N) \right\| \quad (278)$$

$$\leq \sum_{|k_1|, |k_2| \leq \frac{N-1}{2}} e^{-(k_1^2 + k_2^2)t} \|P_{k_1, k_2} - P_{k_1, k_2}^N\|. \quad (279)$$

Next we note

$$\|P_{k_1, k_2} - P_{k_1, k_2}^N\| \leq 2 \|\phi_{k_1, k_2} - \phi_{k_1, k_2}^N\|. \quad (280)$$

It is not hard to see that

$$\left\| \phi_{k_1, k_2} - \overline{\phi_{k_1, k_2}^N} \right\| \leq 2C(|k_1| + |k_2|) \frac{2\pi}{N} \quad (281)$$

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

$$\left\| \sum_{-\frac{N-1}{2} \leq k_1, k_2 \leq \frac{N-1}{2}} e^{-\lambda_{k_1, k_2} t} (P_{k_1, k_2} - P_{k_1, k_2}^N) \right\| \quad (282)$$

$$\leq \sum_{|k_1|, |k_2| \leq \frac{N-1}{2}} e^{-(k_1^2 + k_2^2)t} \cdot 2C(|k_1| + |k_2|) \frac{2\pi}{N} \quad (283)$$

$$= \mathcal{O}(1/N). \quad (284)$$

Where the last claim follows from summability in k_1, k_2 . Thus we have in total indeed established that (262) holds.

In our experiments we then make use of the operators $J_i^{\uparrow\downarrow}$ defined above. The function $f \in L^2(\mathcal{M})$ on the torus from which node features are derived is chosen as $f = \frac{1}{4\pi^2} \sin(\phi) \cos(\theta)$. All networks are asked to predict a scalar signal on the respective graphs.

H. Further Discussion of the setting of coarse-graining Graphs

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.

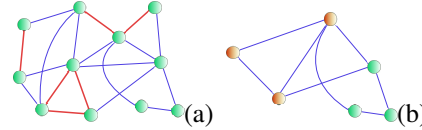


Figure 24. (a) G (strongly connected) clusters in red (b) Coarse grained $\underline{G}$

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 & Vnderghyest, 2018a; Loukas, 2019).

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 25). 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. the discussion below) that

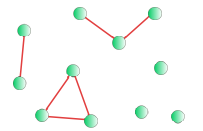


Figure 25. G_{cluster}

$$\|e^{-tL} - J^\uparrow e^{-t\underline{L}} J^\downarrow\| \lesssim 1/w_{\text{high}}^{\min} \text{ for any } t > 0. \quad (285)$$

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 (285) that also $\eta(t) = \|e^{-Lt} - J^\uparrow e^{-\underline{L}t} J^\downarrow\| \rightarrow 0$ 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 $J^\uparrow$.

Following the proof in (Koke, 2026b), we here illustrate

$$\|(L + Id)^{-1} - J^\uparrow(\underline{L} + Id)^{-1} J^\downarrow\| \lesssim 1/\lambda_1(L_{\text{high}}). \quad (286)$$

in this Appendix. Using (Koke, 2026b, Corollary 3.18), this then also implies

$$\|e^{-tL} - J^\uparrow e^{-t\underline{L}} J^\downarrow\| \lesssim 1/w_{\text{high}}^{\min} \text{ for any } t > 0, \quad (287)$$

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

For convenience, we here explicitly restate the definitions leading up to this setting in full clarity and generality:

Definition H.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

$$\underline{W}_{RP} = \sum_{r \in R} \sum_{p \in P} W_{rp}, \quad (288)$$

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

$$\underline{\mathcal{E}} = \{(R, P) \in \underline{\mathcal{G}} \times \underline{\mathcal{G}} : \underline{W}_{RP} > 0\} \quad (289)$$

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

$$\underline{\mu}_R = \sum_{r \in R} \mu_r. \quad (290)$$

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 H.2. Denote by $\mathbb{1}_R$ the vector that has 1 as entries on nodes r belonging to the connected (in G_{high}) 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

$$(J^\downarrow x)_R = \langle \mathbb{1}_R, x \rangle / \underline{\mu}_R. \quad (291)$$

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

$$J^\uparrow u = \sum_R u_R \cdot \mathbb{1}_R; \quad (292)$$

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} .

Slightly extending established results in (Koke, 2026b), we then have the following result:

Theorem H.3. We have

$$\|R_z(L) - J^\uparrow R_z(\underline{L}) J^\downarrow\| = \mathcal{O}\left(\frac{\|L_{\text{reg.}}\|}{\lambda_1(L_{\text{high}})}\right) \quad (293)$$

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

$$\lambda_{\max}(L_{\text{reg.}}) = \|L_{\text{reg.}}\|. \quad (294)$$

We here include the proof of this fact for convenience and self-containedness.

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

$$R_z(L) = (L - zId)^{-1}, \quad (295)$$

$$R_z(L_{high}) = (L_{high} - zId)^{-1} \quad (296)$$

$$R_z(L_{reg.}) = (L_{reg.} - zId)^{-1} \quad (297)$$

the resolvents corresponding to L , L_{high} and $L_{reg.}$ respectively.

Our first goal is establishing that we may write

$$R_z(L) = [Id + R_z(L_{high})L_{reg.}]^{-1} \cdot R_z(L_{high}) \quad (298)$$

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

$$R_z(A + B) - R_z(A) = -R_z(A)BR_z(A + B).” \quad (299)$$

In our case, this translates to

$$R_z(L) - R_z(L_{high}) = -R_z(L_{high})L_{reg.}R_z(L) \quad (300)$$

or equivalently

$$[Id + R_z(L_{high})L_{reg.}] R_z(L) = R_z(L_{high}). \quad (301)$$

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

$$R_z(L) = [Id + R_z(L_{high})L_{reg.}]^{-1} \cdot R_z(L_{high}) \quad (302)$$

as desired.

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

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

$$[Id + R_z(L_{high})L_{reg.}]x = 0. \quad (303)$$

Multiplying with $(L_{high} - zId)$ from the left yields

$$(L_{high} + L_{reg.} - zId)x = 0 \quad (304)$$

which is precisely to say that

$$(L - zId)x = 0 \quad (305)$$

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

$$R_z(L) = [Id + R_z(L_{high})L_{reg.}]^{-1} R_z(L_{high}). \quad (306)$$

Our next step is to establish that

$$R_z(L_{high}) \rightarrow \frac{P_0^{\text{high}}}{-z}, \quad (307)$$

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

$$R_z(L_{high}) \equiv (L_{high} - zId)^{-1} = \sum_{\lambda \in \sigma(L_{high})} \frac{1}{\lambda - z} \cdot P_{\lambda}^{\text{high}}. \quad (308)$$

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

$$\left\| R_z(L_{high}) - \frac{P_0^{high}}{-z} \right\| = \left\| \sum_{0 < \lambda \in \sigma(L_{high})} \frac{1}{\lambda - z} \cdot P_\lambda^{high} \right\|; \quad (309)$$

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

$$\left\| R_z(L_{high}) - \frac{P_0^{high}}{-z} \right\| = \frac{1}{\lambda_1(\Delta_{high}) + |z|}. \quad (310)$$

Here $\lambda_1(L_{high})$ is the first non-zero eigenvalue of (L_{high}) .

Non-zero eigenvalues scale linearly with the weight scale since we have

$$\lambda(S \cdot L) = S \cdot \lambda(L) \quad (311)$$

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

$$\left\| R_z(L_{high}) - \frac{P_0^{high}}{-z} \right\| = \frac{1}{\lambda_1(L_{high}) + |z|} \leq \frac{1}{\lambda_1(L_{high})} \rightarrow 0 \quad (312)$$

as $\lambda_1(L_{high}) \rightarrow \infty$.

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

$$I := \left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} \frac{P_0^{high}}{-z} - [Id + R_z(L_{high}) L_{reg.}]^{-1} R_z(L_{high}) \right\|. \quad (313)$$

To this end we first note that the relation

$$[A + B - zId]^{-1} = [Id + R_z(A)B]^{-1} R_z(A) \quad (314)$$

provided to us by the second resolvent formula, implies

$$[Id + R_z(A)B]^{-1} = Id - B[A + B - zId]^{-1}. \quad (315)$$

Thus we have

$$\left\| [Id + R_z(L_{high}) L_{reg.}]^{-1} \right\| \leq 1 + \|L_{reg.}\| \cdot \|R_z(L)\| \quad (316)$$

$$\leq 1 + \frac{\|L_{reg.}\|}{|z|}. \quad (317)$$

With this, we have

$$\left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} \cdot \frac{P_0^{high}}{-z} - R_z(L) \right\| \quad (318)$$

$$= \left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} \cdot \frac{P_0^{high}}{-z} - [Id + R_z(L_{high})L_{reg.}]^{-1} \cdot R_z(L_{high}) \right\| \quad (319)$$

$$\leq \left\| \frac{P_0^{high}}{-z} \right\| \cdot \left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} - [Id + R_z(L_{high})L_{reg.}]^{-1} \right\| + \left\| \frac{P_0^{high}}{-z} - R_z(L_{high}) \right\| \cdot \left\| [Id + R_z(L_{high})L_{reg.}]^{-1} \right\| \quad (320)$$

$$\leq \frac{1}{|z|} \left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} - [Id + R_z(L_{high})L_{reg.}]^{-1} \right\| + \left(1 + \frac{\|L_{reg.}\|}{|z|} \right) \cdot \frac{1}{\lambda_1(L_{high})}. \quad (321)$$

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

$$\|A^{-1} - B^{-1}\| \leq \frac{\|A^{-1}\| \cdot \|A^{-1}C\|}{1 - \|A^{-1}C\|}. \quad (322)$$

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

$$\left\| \left[Id + P_0^{high}/(-z) \cdot L_{reg.} \right]^{-1} - [Id + R_z(L_{high})L_{reg.}]^{-1} \right\| \quad (323)$$

$$\leq \frac{(1 + \|L_{reg.}\|/|z|)^2 \cdot \|L_{reg.}\| \cdot \left\| \frac{P_0^{high}}{-z} - R_z(L_{high}) \right\|}{1 - (1 + \|L_{reg.}\|/|z|) \cdot \|L_{reg.}\| \cdot \left\| \frac{P_0^{high}}{-z} - R_z(L_{high}) \right\|} \quad (324)$$

For S_{high} sufficiently large, we have

$$\left\| -P_0^{high}/z - R_z(L_{high}) \right\| \leq \frac{1}{2(1 + \|L_{reg.}\|/|z|)} \quad (325)$$

so that we may estimate

$$\left\| \left[Id + L_{reg.} \frac{P_0^{high}}{-z} \right]^{-1} - [Id + L_{reg.} R_z(L_{high})]^{-1} \right\| \quad (326)$$

$$\leq 2 \cdot (1 + \|L_{reg.}\|) \cdot \left\| \frac{P_0^{high}}{-z} - R_z(L_{high}) \right\| \quad (327)$$

$$= 2 \frac{1 + \|L_{reg.}\|/|z|}{\lambda_1(L_{high})} \quad (328)$$

Thus we have now established

$$\left\| \left[Id + \frac{P_0^{high}}{-z} L_{reg.} \right]^{-1} \cdot \frac{P_0^{high}}{-z} - R_z(L) \right\| = \mathcal{O} \left(\frac{\|L_{reg.}\|}{\lambda_1(L_{high})} \right). \quad (329)$$

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

$$\left[-zId + P_0^{high} L_{reg.} \right]^{-1} P_0^{high} = J^\uparrow R_z(L) J^\downarrow, \quad (330)$$

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

$$J^\uparrow \cdot J^\downarrow = P_0^{high} \quad (331)$$

and

$$J^\downarrow \cdot J^\uparrow = Id_{\underline{G}}. \quad (332)$$

Indeed, the relation (331) follows from the fact that the eigenspace corresponding to the eigenvalue zero is spanned by the vectors $\{\mathbb{1}_R\}_R$, with $\{R\}$ the connected components of G_{high} . Equation (332) follows from the fact that

$$\langle \mathbb{1}_R, \mathbb{1}_R \rangle = \mu_R. \quad (333)$$

With this we have

$$\left[Id + P_0^{high} L_{\text{reg.}} \right]^{-1} P_0^{high} = \left[Id + J^\uparrow J^\downarrow L_{\text{reg.}} \right]^{-1} J^\uparrow J^\downarrow. \quad (334)$$

To proceed, set

$$\underline{x} := F^\downarrow x \quad (335)$$

and

$$\mathcal{X} = \left[P_0^{high} L_{\text{reg.}} - z Id \right]^{-1} P_0^{high} x. \quad (336)$$

Then

$$\left[P_0^{high} L_{\text{reg.}} - z Id \right] \mathcal{X} = P_0^{high} x \quad (337)$$

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

$$J^\uparrow J^\downarrow (L_{\text{reg.}} - z Id) J^\uparrow J^\downarrow \mathcal{X} = J^\uparrow J^\downarrow x. \quad (338)$$

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

$$J^\downarrow (L_{\text{reg.}} - z Id) J^\uparrow J^\downarrow \mathcal{X} = J^\downarrow x. \quad (339)$$

Thus we have

$$(J^\downarrow L_{\text{reg.}} J^\uparrow - z Id) J^\uparrow J^\downarrow \mathcal{X} = J^\downarrow x. \quad (340)$$

This – in turn – implies

$$J^\uparrow J^\downarrow \mathcal{X} = \left[J^\downarrow L_{\text{reg.}} J^\uparrow - z Id \right]^{-1} J^\downarrow x. \quad (341)$$

Using

$$P_0^{high} \mathcal{X} = \mathcal{X}, \quad (342)$$

we then have

$$\mathcal{X} = J^\uparrow \left[J^\downarrow L_{\text{reg.}} J^\uparrow - z Id \right]^{-1} J^\downarrow x. \quad (343)$$

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

As a corollary, one finds

Corollary H.4. *We have*

$$R_z(L)^k \rightarrow J^\uparrow R^k(\underline{L}) J^\downarrow \quad (344)$$

Proof. This follows directly from the fact that

$$J^\downarrow J^\uparrow = Id_{\underline{G}}. \quad (345)$$

$\square$

I. Computational cost analysis

To evaluate the practical efficiency of our proposed framework, we conducted a thorough empirical investigation of the computational costs associated with constructing and operating on a representative propagation matrix—specifically, the resolvent formulation introduced in Def, 6.1.

I.1. Computational Efficiency on Small-Scale Graphs (QM7)

We first established a baseline for resource consumption during model training on the QM7 molecular dataset. To quantify computational overhead, we monitored key performance metrics throughout training:

- **Peak GPU Memory Allocation** was tracked using `torch.cuda.memory.max_memory_allocated`.
- **Training Time per Epoch** was measured as the wall-clock time required to complete a single forward and backward pass across all training samples.

Table 6. Computational overhead on small graphs

Model	Max Memory [MiB]	Time/Epoch [s]
ChebNet	230.8±0.8	0.8±0.1
GAT	248.0±0.5	1.0±0.1
Resolvent	184.8±0.1	0.7±0.1

As evident from Table 6, our method has significantly less overhead than standard GNN baselines.

I.2. Scalability to Large-Scale Graphs

To assess how our approach scales to larger, more complex topologies, we extended our evaluation to datasets with significantly higher node counts, namely `ms_academic` (> 18,000 nodes) and `pubmed` (> 19,000 nodes).

Verification of effective sparsity: We first experimentally verify the claim made towards the end of Section 7.1 that the propagation matrix is effectively sparse.

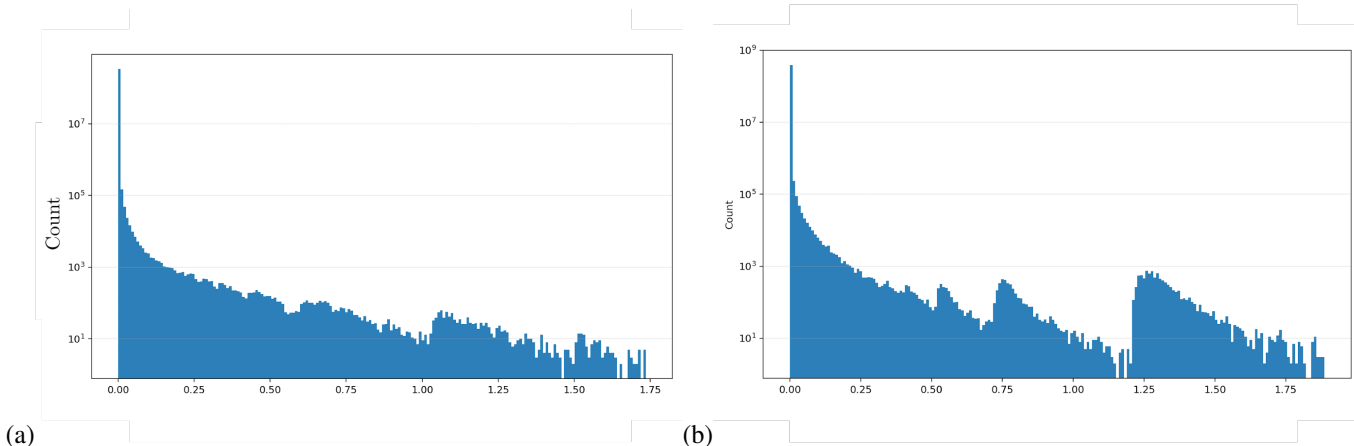


Figure 26. Histogram of $|R_{ij}|$ for resolvent matrix R on (a) `ms_academic` and (b) `pubmed`.

As is evident from Figure 26, the overwhelming majority of entries R_{ij} in the resolvent propagation matrix R are indeed effectively zero. Indeed, we list results for small cutoff values $\epsilon \geq |R_{ij}|$ in Table 7 below.

Table 7. Effective sparsity structure

Dataset	$ R_{ij} \geq 0.05$		$ R_{ij} \geq 0.01$	
	Total number [#]	Percentage [%]	Total number [#]	Percentage [%]
PubMed	121,053	0.03%	500,571	0.13%
MS-Academic	56,096	0.02%	257,475	0.07%

Computational overhead analysis Using the identical logging infrastructure and metrics as in the QM7 experiments, we tracked the scaling behavior of memory allocation and epoch duration.

Table 8. Computational overhead on large graphs

Model	Max Memory [MiB]	Time/Epoch [ms]
ChebNet	551.0 $\pm$ 9.0	107.14 $\pm$ 1.21
GAT	2473.6 $\pm$ 10.4	270.78 $\pm$ 101.25
Resolvent	168.9 $\pm$ 0.9	35.86 $\pm$ 0.19

Our empirical results validate the theoretical assertions presented in our scalability discussion at the end of Section 7.1.