GRAPH NETWORKS STRUGGLE WITH VARIABLE SCALE

Christian Koke[1,2]**, Yuesong Shen**[1,2]**, Abhishek Saroha**[1,2]**, Marvin Eisenberger**[1]
Bastian Rieck[3], **Michael Bronstein**[4], **Daniel Cremers**[1,2]
[1]Technical University Munich, [2]Munich Center for Machine Learning,
[3]University of Fribourg, [4]University of Oxford

ABSTRACT

Standard graph neural networks assign vastly different latent embeddings to graphs describing the same object at different resolution scales. This precludes consistency in applications and prevents generalization between scales as would fundamentally be needed e.g. in AI4Science. We uncover the underlying obstruction, investigate its origin and show how to overcome it by modifying the message passing paradigm.

1 INTRODUTION

While the real-world impact of graph neural networks is undeniable, their developmental pipeline arguably still remains somewhat decoupled from the challenges faced in real-world settings. Beyond well-known limitations of standard datasets (Wu et al., 2021; Ying et al., 2018; Hu et al., 2020; Glavatskikh et al., 2019; Platonov et al., 2023), we here want to shed light on an additional issue typically faced by GNNs, so far overlooked in the literature: Standard graph learning methods are unable to consistently incorporate (resolution-)scale information within graphs.

To understand the significance of this problem, consider two graphs representing the same surface at different resolution scales; say as obtained utilizing 3D scanners with distinct imaging resolutions. As both graphs describe the same underlying object, a GNN-based classification architecture should be able to correctly classify the scanned surface regardless of which graph is used to represent it. Similarly, a social network might be accessible during training at the level of individual interacting users. During inference, a user might however only have information about the same social network at a much coarser scale; say at the aggregate level of interacting communities (or organisations) as opposed to potentially proprietary or protected information about detailed interactions between individual users. Nevertheless, when querying the GNN with a graph at this coarser scale, we still hope for accurate predictions generated by the model.

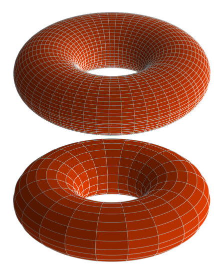

Figure 1: Torus (two resolutions)

Maybe most fundamentally, graphs are also used in AI4Science, where they model the pairwise interactions between the constituents of a given physical system (e.g. small volume elements in fluid dynamics (Sanchez-Gonzalez et al., 2020)). In this scientific setting, training data is generically available only at a coarse scale, as the generation of fine-detail training data even for modestly-sized systems is prohibitively expensive (Feynman, 1981). Thus developing machine learning models that can be trained on coarse-scale training data while still being able to generalize to more complex higher-resolution systems during inference is a fundamental goal in this domain (Kazeev et al., 2024).

2 THE FAILURE MODE: INCONSISTENT INCORPORATION OF SCALE

To show that standard graph learning methods fail to achieve this and are in fact unable to consistently integrate varying scales we make use of the QM7 dataset (Rupp et al., 2012) (c.f. Appendices G.2 through G.4 for additional experimental settings). This dataset consists of organic molecules containing both hydrogen and heavy atoms. The 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.

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. Instead, only an aggregate description is used and each nucleus is described by a single node. In order to test the ability of GNNs to do inference on a scale different from which they were trained on, we additionally also consider a version of QM7 where we lower the resolution scale even further: Here we aggregate each heavy atomic core additionally together with its surrounding (single-proton) hydrogen atoms into super-nodes. Appendix G.1 provides exact details. We might interpret this $\text{QM7}_{\text{coarse}}$ dataset as a model for data obtained from a resolution-limited observation process unable to resolve positions of individual (small) hydrogen atoms and only providing information about how many hydrogen atoms are bound to a given heavy atom.

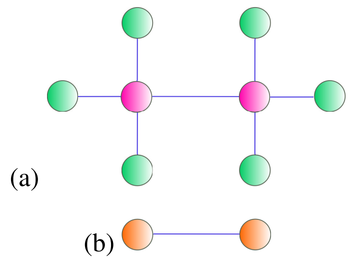

Figure 2: (a) Original graph G corresponding to the Ethane molecule with Carbon in purple and Hydrogen in green (b) Coarse grained $\underline{G}$ with aggregate Carbon-Hydrogen super-nodes in orange

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

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

Table 1 collects results. Mean-absolute-errors (MAEs) made during inference increase significantly, when going from a same-resolution setting to a cross-resolution setting. None of the considered standard architectures are able to consistently handle more than one scale. We can trace this back to the latent embeddings $\{F\}$ and $\{\underline{F}\}$ generated for original graphs $\{G\}$ and coarsified graphs $\{\underline{G}\}$: For models of Table 1 on average $10 \lesssim \|F - \underline{F}\| \lesssim 10^4$ (c.f. also Fig. 4 below). Thus latent embeddings generated for graphs describing the same object on varying resolutions are significantly different. We especially note, that models trained on the coarse scale do not generalize to fine-scale graphs. Note that in practice, this problem may also not be remedied by augmenting the training set, as we have no way of generating faithful high-resolution descriptions given only lower resolution graphs.

3 Identifying the Problem: Standard GNNs are not continuous

Within the coarse graphs $\{\underline{G}\}$ of $\text{QM7}_{\text{coarse}}$, we have fused hydrogen atoms onto the respective nearest heavy atoms. We can think of the resulting graph as being the limit of a procedure where hydrogen atoms are moved out of equilibrium towards their respective nearest heavy atom. The limit graph is then a coarse grained graph where hydrogen atoms have been captured by the respective heavy atoms.

If standard GNN architectures would act as continuous maps from the space of graphs to the chosen latent space, then the convergence of this graph modification process 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.** In Figure 4, we thus compare 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 $\omega \geqslant 1$ (i.e. $\text{dist}_{\text{new}} = \text{dist}_{\text{equilib.}}/\omega$), but have not yet completely arrived at the positions of nearest heavy atoms.

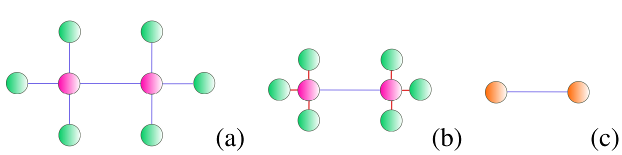

Figure 3: Collapsing Procedure visualized

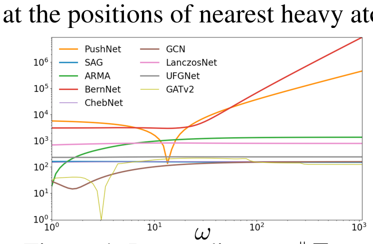

Figure 4: Latent distance $\|F_\omega - \underline{F}\|$

Figure 4 shows however, that latent embeddings do *not* converge ($\|F_\omega - \underline{F}\| \nrightarrow 0$). Thus GNNs cannot be considered continuous and hence may map similar graphs to vastly different latent embeddings.

4 Understanding the Problem: Disconnected propagation schemes

We can understand the underlying reason for this discontinuity by exemplarily investigating GCN (Kipf & Welling, 2017) (c.f. Appendix B for other methods): 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 $X \mapsto \hat{A}XW$. Here $W \in \mathbb{R}^{F \times F}$ facilitates channel mixing, while information flow over the graph is implemented via the *renormalized* adjacency matrix $\hat{A} \in \mathbb{R}^{N \times N}$; given as $\hat{A}_{ij} \sim A_{ij}/\sqrt{d_i d_j}$ (with degrees d_i). As we move hydrogen (H) atoms towards heavy atoms ($|\vec{x}_{\text{H}} - \vec{x}_{\text{heavy}}| \to 0$), corresponding edge weights $A_{\text{H,heavy}} = 1 \cdot Z_{\text{heavy}} \cdot |\vec{x}_{\text{H}} - \vec{x}_{\text{heavy}}|^{-1}$ 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) between heavy atoms remain constant, however, the *renormalized* entries $\hat{A}_{\text{heavy,heavy}}$ tend to zero instead.

Thus 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 ($\hat{A}_{\text{heavy,heavy}} \to 0$). Information is only propagated along a severely disconnected effective limit graph (dissected into distinct connected components; Fig 5 (a)) and not along the true limit graph $\underline{G}$ (Fig. 5 (b)).

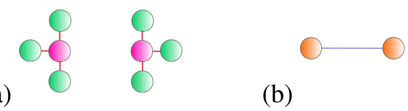
(a) (b)

Figure 5: (a) Effective propagation graph vs (b) true lower-resolution graph $\underline{G}$

Since the information-flows over the graphs $G_\omega, \underline{G}$ are vastly different, also the latent embeddings $F_\omega, \underline{F}$ that are being generated for the two respective graphs are vastly different.

5 Solving the Problem: GNNs with global Laplacian propagation

To build architectures that will instead be continuous in the setting above, let us formalize rigorously, in which sense the sequence of graphs G_ω approaches the limit graph $\underline{G}$. We first observe that, when moving hydrogen atoms out of equilibrium, we are significantly increasing certain weights ($A_{\text{H,heavy}} \sim |\vec{x}_{\text{H}} - \vec{x}_{\text{heavy}}|^{-1} \sim \omega \to \infty$). From a 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 and 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 we establish rigorously in Appendix C we then have

$$\eta_\omega(t) := \|e^{-tL_\omega} - J^{\uparrow} e^{-t\underline{L}} J^{\downarrow}\| \to 0 \quad \text{for any fixed } t > 0 \text{ as } \omega \to \infty, \tag{1}$$

Here $L_\omega, \underline{L}$ are the Laplacians of the respective graphs $G_\omega, \underline{G}$. The matrices $J^{\downarrow,\uparrow}$ linerarly interpolate between the graphs G_ω and $\underline{G}$ (of different sizes): $J^{\downarrow}$ assigns the average over strongly connected clusters to the super-node representing this cluster in $\underline{G}$. $J^{\uparrow}$ is its adjoint ($J^{\uparrow} = [J^{\downarrow}]^{\intercal}$).

We might interpret (1) as telling us that applying the matrix e^{-tL_ω} is more and more the same as projecting to $\underline{G}$ via $J^{\downarrow}$, applying the matrix $e^{-t\underline{L}}$ there and interpolating back up via $J^{\uparrow}$. Thus, while the propagation rule $X \mapsto \hat{A}_\omega XW$ is insufficient and leads to disconnected limit graphs, propagating as $X \mapsto e^{-tL_\omega} XW$, *does* facilitate contact and similarity between information flows over G_ω and $\underline{G}$.

More generally, suppose we have for each time $t \geqslant 0$ individually that $\|e^{-Lt} - J^{\uparrow} e^{-t\underline{L}t} J^{\downarrow}| < \delta$. 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 $\|\psi(L_\omega) - J^{\uparrow}\psi(\underline{L})J^{\downarrow}\| \leqslant (\sum_k |a_k|) \cdot \delta$ by a triangle-inequality argument. Thus we can still guarantee that for large ω the propagation $X \mapsto \psi(L_\omega)XW$ over G_ω is approximately the same as the effective propagation $X \mapsto [J^{\uparrow}\psi(\underline{L})J^{\downarrow}]XW$ over $\underline{G}$. Following this idea, we define:

Definition 5.1. Let $\hat{\psi}$ be a bounded (generalized) function defined on $[0, \infty)$. A **Global Laplacian Propagation Matrix** ψ(L) is any matrix arising as $\psi(L) := \int_0^\infty e^{-tL}\hat{\psi}(t)dt$.

Appendix D contains details. Allowing *generalized* functions means we e.g. allow Dirac distributions $\hat{\psi}_{\delta_{t_k}}(t) := \delta(t - t_k)$; leading to **exponential** matrices $\psi_k(L) = \int_0^\infty \delta(t - t_k)e^{-tL}dt = e^{-t_k L}$. Choosing e.g. $\hat{\psi}_k := (-t)^{k-1}e^{-\lambda t}$ instead yields powers of **resolvents** $\psi_k(L) = [(zId + L)^{-1}]^k$.

In Appendix F.1 we then prove the following result; implying $\|F_\omega - \underline{F}\| \to 0$ as $\eta_\omega(t) \to 0$ in (1):

Theorem 5.2. Let $\{\hat{\psi}_k\}_k$ be a collection of bounded generalized functions. Consider a network for which – when deployed on a graph G – the layer-wise update rule is implemented as $X \mapsto \sum_k \psi_k(L)XW_k$, with L the Laplacian of G and the W_ks implementing channel mixing.
Then in the setting of (1) we have $\|F_\omega - \underline{F}\| \lesssim \max_k \left\{\int_0^\infty |\hat{\psi}_k(t)|\eta_\omega(t)dt\right\} \to 0$.

6 Verification: Global Laplacian Propagation solves the problem

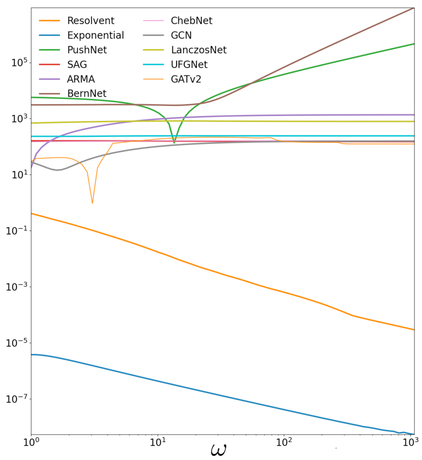

Figure 6: Latent distance $\|F_\omega - \underline{F}\|$

Table 2: Regression using high- and low-resolution QM7

Mean Absolute Error (↓) on QM7 [kcal/mol]				
Training	**High Resolution**		**Low Resolution**	
Inference	**Low Resolution**	High Resolution	Low Resolution	**High Resolution**
GCN	$125.34_{\pm 2.47}$	$63.17_{\pm 0.92}$	$67.75_{\pm 3.73}$	$380.51_{\pm 30.33}$
PushNet	$2442.59_{\pm 303.27}$	$60.94_{\pm 1.83}$	$69.25_{\pm 3.11}$	$124.08_{\pm 3.94}$
Resolvent	$16.54_{\pm 3.01}$	$16.53_{\pm 3.03}$	$15.79_{\pm 0.98}$	$13.80_{\pm 1.34}$
Exponential	$16.37_{\pm 1.71}$	$16.36_{\pm 2.16}$	$16.25_{\pm 1.41}$	$16.25_{\pm 1.41}$

Theorem 5.2 implies that networks employing global Laplacian propagation schemes are indeed continuous as maps from the space of graphs into their latent spaces. To numerically verify this, we repeat the experiment of Section 3 for two models belonging to this category (using resolvent and exponential matrices; c.f. Section 5). As is evident from Fig. 6, latent embeddings generated by models employing global Laplacian propagation *do* converge.

In Section 3 we had identified lack of continuity as the obstruction to generalizing between scales. Since graph neural networks based on global Laplacian propagation *are* continuous (and hence map similar graphs to similar latent embeddings), we hence expect them to generalize between resolution scales as well. To verify this, we here repeat the experiment of Section 2 again with these networks.

Table 6 details that MAEs of GNNs based on global Laplacian propagation schemes (using either exponential or resolvent matrices) do not increase when going from a same- to a cross-resolution setting. Comparing with Table 1, we see that in cross-resolution settings MAEs of methods employing global Laplacian propagation schemes are lower than those of standard graph learning methods by factors of order 10^1 to 10^2: The methods developed in Section 5 indeed do generalize between scales.

We can further understand this generalization ability using Theorem 5.2: Exemplarily considering exponential propagation matrices (c.f. Section 5) we have that $\int_0^\infty |\hat{\psi}_k(t)|\eta_\omega(t)dt = \int_0^\infty \delta(t-t_k)\eta_\omega(t)dt = \eta(t_k)$. Choosing $t_k = k$ (as for the architecture investigated in Table 2; c.f. details in Appendix G.1), we thus have $\|F - \underline{F}\| \lesssim \max_{k\geqslant 1}|\eta(k)|$. When investigating the differences $\eta(t) = \|e^{-tL} - J^\uparrow e^{-t\underline{L}}J^\downarrow\|$ of diffusion flows, we find that $\eta(t)$ drops to zero fast, as exemplarily plotted in Fig. 7 for the first few molecules of QM7. In particular $\eta(k)|_{k\geqslant 1} \lesssim 10^{-2}$. Using this as an upper bound in Theorem 5.2 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.

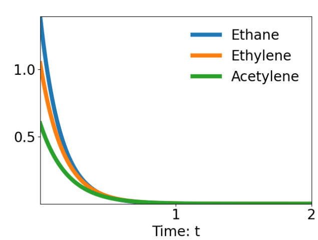

Figure 7: $\|e^{-Lt} - J^\uparrow e^{-t\underline{L}}J^\downarrow\|$

7 Summary

In this paper, we discussed the inability of existing graph learning methods to incorporate multiple scales. We found the underlying obstruction to be a lack of continuity when GNNs are considered as maps from the space of graphs to their latent space. We derived how to build continuous models instead and showed that these models can indeed consistently incorporate varying scales.

REFERENCES

Wolfgang Arendt. Approximation of degenerate semigroups. *Taiwanese Journal of Mathematics*, 5 (2):279 – 295, 2001. doi: 10.11650/twjm/1500407337. URL https://doi.org/10.11650/twjm/1500407337.

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

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

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

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

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

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

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

Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1981. doi: 10.1007/BF02650179.

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

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

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

Marta Glavatskikh, Jules Leguy, Gilles Hunault, Thomas Cauchy, and Benoit Da Mota. Dataset's chemical diversity limits the generalizability of machine learning predictions. *Journal of Cheminformatics*, 11, 2019. URL https://api.semanticscholar.org/CorpusID:207960025.

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

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

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

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

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

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

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

Nikita Kazeev, Eleonore Vissol-Gaudin, Mengyi Chen, Isabelle Guyon, and Andrey E Ustyuzhanin. Machine learning multiscale processes, 2024. URL https://openreview.net/forum?id=uAtjhZW1QV.

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

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

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

Christian Koke. Strong connectivity in graphs: Norm resolvent convergence to effective descriptions, 2025.

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

Christian Koke and Gitta Kutyniok. 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.

Christian Koke, Abhishek Saroha, Yuesong Shen, Marvin Eisenberger, and Daniel Cremers. Resolvnet: A graph convolutional network with multi-scale consistency. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023. URL https://arxiv.org/abs/2310.00431.

Christian Koke, Abhishek Saroha, Yuesong Shen, Marvin Eisenberger, Michael M. Bronstein, and Daniel Cremers. 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.

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

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

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: are we really making progress? *ArXiv*, abs/2302.11640, 2023. URL https://api.semanticscholar.org/CorpusID:257102689.

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

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

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. *ArXiv*, abs/2002.09405, 2020. URL https://api.semanticscholar.org/CorpusID:211252550.

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

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

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

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. URL https://api.semanticscholar.org/CorpusID:46949657.

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

A Background: (Spectral) Convolutional Networks on Graphs

The architecture proposed in Section 5 (c.f. Theorem 5.2) can be thought of as a particular type of spectral convolutional network. We hence discuss this type of architecture here in more detail:

A.1 GRAPHS AND THEIR FUNDAMENTAL PROPERTIES

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

Feature spaces: Given F-dimensional node features on a graph with $N = |\mathcal{G}|$ nodes, we collect individual scalar node-signals $x \in \mathbb{R}^N$ into a feature matrix X of dimension $N \times F$. Taking node weights into account, we equip the space of such signals with an inner-product according to $\langle X, Y \rangle = \text{Tr}(X^\intercal MY) = \sum_{i=1}^N \sum_{j=1}^F (\overline{X}_{ij} Y_{ij})\mu_i$ with $M = \text{diag}(\{\mu_i\})$ the diagonal matrix of node-weights. Associated to this inner product is the feature norm $\|X\| = (\langle X, X \rangle)^{\frac{1}{2}}$.

Graph Laplacians: Information about the geometry of a graph is encapsulated into the set of edge weights. From this information, various characteristic matrix operators encoding the geometry of the underlying graph may be constructed. Spectral graph neural networks are typically based on some choice of (positive semi-definite) graph Laplacian L (Defferrard et al., 2016; He et al., 2021; 2022). Most important to us is the un-normalized (in-degree) graph Laplacian $L = M^{-1}(D - A)$, due to its intrinsic relation to heat-diffusion on graphs and its ability to capture, disentangle an encode information on graph structure into its its eigenvalue structure (Chung, 1997). Here A is the (weighted) adjacency matrix, D is the diagonal (in-)degree matrix and M is the matrix of node-weights defined above. The 'size' of such a characteristic operator L is measured in spectral norm: $\|L\| = \sup_{\|x\|=1} \|Lx\|$ with $x \in \mathbb{R}^N$ a scalar graph signal.

A.2 SPECTRAL CONVOLUTIONAL FILTERS

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

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

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

A.3 SPECTRAL GRAPH CONVOLUTIONAL NETWORKS:

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

$$X^\ell_{i:} = \rho\left(\sum_{j=1}^{F_{\ell-1}} h^\ell_{\theta_{ij}}(L)(X^{\ell-1}_{j:}) + B^\ell_{i:}\right) \quad (2) \qquad \Leftrightarrow \qquad X^\ell = \rho\left(\sum_{i \in I} \psi_i(L) \cdot X^{\ell-1} \cdot W^\ell_i + B^\ell\right) \quad (3)$$

with biases $B^\ell \in \mathbb{R}^{N \times F_\ell}$ ($B_{:j} = b_j \cdot \mathbb{1}_G$) and weight matrices $W^\ell_i \in \mathbb{R}^{F_{\ell-1} \times F_\ell}$. We here consider activation functions ρ satisfying $\rho(0) = 0$ and $|\rho(a) - \rho(b)| \leqslant |a - b|$ such as e.g. (leaky-)ReLu. The scalar (2) and matrix (3) viewpoints are connected via the identity $h_{\theta_{ij}}(L) \equiv \sum_k (W_k)_{ij} \psi_k(L)$. With basis functions $\Psi = \{\psi_i\}_{i \in I}$, weights $\mathscr{W}$ and biases $\mathscr{B}$, we denote the output of a graph neural network based on the operator L and applied to the node feature matrix X as $\Phi = \Phi_{\mathscr{W}, \mathscr{B}, \Psi}(L, X)$.

B EFFECTIVE PROPAGATION SCHEMES

For definiteness, we here discuss limit-propagation schemes in the setting where **edge-weights** are large. A discussion for high-connectivity in the sense of large cliques is also possible and proceeds analogously.

In this section, we then take up again the setting of Section 4. We reformulate this setting here in a slightly modified language, that is more adapted to discussing effective propagation schemes of standard architectures:

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

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

and the set with small weights is given by:

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

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

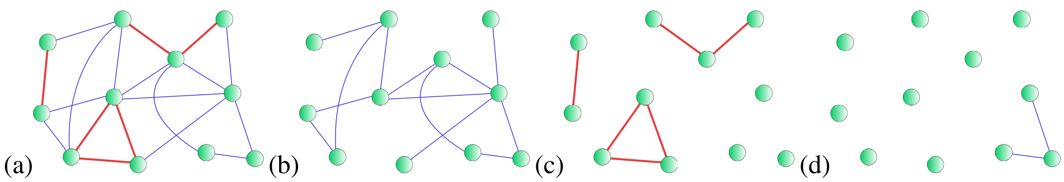

Figure 8: (a) Graph G with $\mathcal{E}_{\text{reg.}}$ (blue) & $\mathcal{E}_{\text{high}}$ (red); (b) $G_{\text{reg.}}$; (c) G_{high}; (d) $G_{\text{reg., exclusive}}$

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. 8).
We also introduce the set of edges $\mathcal{E}_{\text{reg., exclusive}} := \{(i,j) \in \mathcal{E}_{\text{reg.}} | \forall k \in \mathcal{G} : (i,k) \notin \mathcal{E}_{\text{high}} \,\&\, (k,j) \notin \mathcal{E}_{\text{high}}\}$ connecting nodes that do not have an incident edge in $\mathcal{E}_{\text{high}}$. A corresponding example-graph $G_{\text{reg., exclusive}}$ is depicted in Fig. 8 (d).

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

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

B.1 Spectral Convolutional Filters

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

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

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

B.1.1 Need for Normalization

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

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

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

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

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

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

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

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

$$\sigma\left(g_\theta(T)\right) = \{g_\theta(\lambda) : \lambda \in \sigma(T)\}, \tag{4}$$

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

$$|\lambda_{\max}| = \|T\|. \tag{5}$$

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

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

Thus in view of (4) and (5) we have for $\|T\|$ sufficiently large, that

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

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

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

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

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

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

B.1.2 Spectral Normalizations

As discussed in the previous Section B.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

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

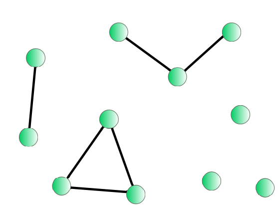

Figure 9: Limit graph corresponding to Fig 8 for spectral normalization

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. 9) only, as Theorem B.1 below establishes:

Theorem B.1. With
$$T = \frac{1}{\lambda_{\max}(\Delta)}\Delta,$$
and the scale decomposition as above we have that
$$\left\| T - \frac{1}{\lambda_{\max}(\Delta_{\text{high}})}\Delta_{\text{high}} \right\| = \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right) \tag{6}$$
for $S_{\text{high}} \gg S_{\text{reg.}}$.

Proof. For convenience in notation, let us write
$$T_{\text{high}} = \frac{1}{\lambda_{\max}(\Delta_{\text{high}})}\Delta_{\text{high}}$$
and similarly
$$T_{\text{reg.}} = \frac{1}{\lambda_{\max}(\Delta_{\text{reg.}})}\Delta_{\text{reg.}}.$$
We may write
$$\Delta = \Delta_{\text{high}} + \Delta_{\text{reg.}},$$
which we may rewrite as
$$\Delta = \lambda_{\max}(\Delta_{\text{high}}) \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}}\right). \tag{7}$$
Let us consider the equivalent expression
$$\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta = T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}}. \tag{8}$$
We next note that
$$\lambda_{\max}\left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta\right) = \frac{\lambda_{\max}(\Delta)}{\lambda_{\max}(\Delta_{\text{high}})}. \tag{9}$$
and
$$\lambda_{\max}\left(T_{\text{high}}\right) = 1$$
since the operation of taking eigenvalues of operators is multiplicative in the sense of
$$\lambda_{\max}(|a| \cdot T) = |a| \cdot \lambda_{\max}(T)$$
for non-negative $|a| \geqslant 0$.

Since the right-hand-side of (8) constitutes an analytic perturbation of T_{high}, we may apply analytic perturbation theory (c.f. e.g. Kato (1976) for an extensive discussion) to this problem. With this (together with $\|T_{\text{high}}\| = 1$) we find
$$\lambda_{\max}\left(\frac{1}{\lambda_{\max}(\Delta_{\text{high}})} \cdot \Delta\right) = 1 + \mathcal{O}\left(\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})}\right). \tag{10}$$
Using (9) and the fact that
$$\frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \propto \frac{S_{\text{reg.}}}{S_{\text{high}}}, \tag{11}$$
we thus have
$$\frac{\lambda_{\max}\left(\Delta\right)}{\lambda_{\max}(\Delta_{\text{high}})} = 1 + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right).$$
Since for small ϵ, we also have
$$\frac{1}{1+\epsilon} = 1 + \mathcal{O}(\epsilon),$$
the relation (11) also implies
$$\frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}\left(\Delta\right)} = 1 + \mathcal{O}\left(\frac{S_{\text{reg.}}}{S_{\text{high}}}\right).$$

Multiplying (7) with $1/\lambda_{\max}(\Delta)$ yields

$$T = \frac{\lambda_{\max}(\Delta_{\text{high}})}{\lambda_{\max}(\Delta)} \cdot \left(T_{\text{high}} + \frac{\lambda_{\max}(\Delta_{\text{reg.}})}{\lambda_{\max}(\Delta_{\text{high}})} \cdot T_{\text{reg.}} \right). \tag{12}$$

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

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

for sufficiently large S_{high}, relation (12) implies

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

as desired.

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

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

for self-adjoint operators A and B and their respective k^{th} eigenvalues ordered by magnitude. On a graph with N nodes, we clearly have $\lambda_{\max} = \lambda_N$ for eigenvalues of (rescaled) graph Laplacians, since all such eigenvalues are non-negative. This implies for the difference $|1 - \lambda_{\max}(\Delta)/\lambda_{\max}(\Delta_{\text{high}})|$ arising in (10) that explicitly

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

This in turn can then be used to provide a quantitative bound in (6). Since we are only interested in the qualitative behaviour for $S_{\text{high}} \gg S_{\text{reg.}}$, we shall however not pursue this further.

□

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

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

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

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

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

for Hermitian A, B. Since we have

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

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

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

for arbitrary hermitian A, B.

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

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

With this we find

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

□

B.1.3 Symmetric Normalizations

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

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

In the setting $S_{\text{high}} \gg S_{\text{reg.}}$ we are considering, this leads to an effective feature propagation along edges in $\mathcal{E}_{\text{high}}$ and $\mathcal{E}_{\text{low, exclusive}}$ (c.f. also Fig. 10) only, as Theorem B.3 below establishes:

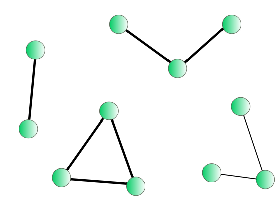

Figure 10: Limit graph corresponding to Fig 8 for symmetric normalization

Theorem B.3. With

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

and the scale decomposition as introduced above, we have that

$$\left\|T - \left(Id - D_{\text{high}}^{-\frac{1}{2}}W_{\text{high}}D_{\text{high}}^{-\frac{1}{2}} - D_{\text{reg.}}^{-\frac{1}{2}}W_{\text{low, exclusive}}D_{\text{reg.}}^{-\frac{1}{2}}\right)\right\| = \mathcal{O}\left(\sqrt{\frac{S_{\text{reg.}}}{S_{\text{high}}}}\right) \tag{13}$$

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

Proof. We first note that instead of (13), we may equivalently establish

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

We have

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

With this, we may write

$$D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = D^{-\frac{1}{2}}W_{\text{high}}D^{-\frac{1}{2}} + D^{-\frac{1}{2}}W_{\text{reg.}}D^{-\frac{1}{2}}. \tag{14}$$

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

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

Let us use the notation

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

We then find

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

Using the Taylor expansion

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

we thus have

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

Since we have

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

this yields

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

Thus let us turn towards the second summand on the right-hand-side of (14). We have

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

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

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

Since

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

we have

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

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

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

Thus in total we have established

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

which was to be established.

□

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

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

with

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

and

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

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

Theorem B.4. With

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

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

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

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

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

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

The difference to the result of Theorem B.3 is thus that applicability of the limit propagation scheme of Fig. 10 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 B.3 – we first decompose T:

$$\begin{aligned} \tilde{D}^{-\frac{1}{2}}\tilde{W}\tilde{D}^{-\frac{1}{2}} &= \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} Id \tilde{D}^{-\frac{1}{2}} \\ &= \tilde{D}^{-\frac{1}{2}} W_{\text{high}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-\frac{1}{2}} W_{\text{reg.}} \tilde{D}^{-\frac{1}{2}} + \tilde{D}^{-1} \end{aligned} \tag{15}$$

For the first term, we note

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

We then find

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

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

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

This implies

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

Next we turn to the second summand in (15):

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

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

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

Since

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

we have

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

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

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

Finally we note for the third term on the right-hand-side of (15) that

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

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

In total we thus have found

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

which was to be proved. □

B.2 Spatial Convolutional Filters

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

$$X_i^{\ell+1}=\gamma\left(X_i^\ell,\coprod_{j\in\mathcal{N}(i)}\phi\left(X_i^\ell,X_j^\ell,w_{ij}\right)\right). \tag{16}$$

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

Before we discuss corresponding limit-propagation schemes, we first establish that MPNNs are not able to reproduce the limit propagation scheme ofFigure 5 (b) and are thus not stable to scale transitions and topological perturbations.

B.2.1 Scale-Sensitivity of Message Passing Neural Networks

Here we establish that message passing networks (as defined in (16) above) are unable to emulate a limit propagation scheme similar to the one in Figure 5 (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. 11). We assume all node-weights to be equal to one ($\mu_i = 1$ for $i = 1, 2, 3$) and edge weights

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

as well as

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

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

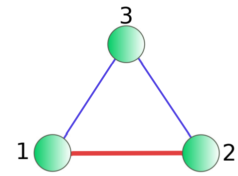

Figure 11: Three node Graph G with on large weight $w_{12} \gg 1$.

Given states $\{X_1^\ell, X_2^\ell, X_3^\ell\}$ in layer ℓ, a limit propagation scheme as in Figure 5 (b) would require the updated feature vector of node 3 to be given by

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

However, the actual updated feature at node 3 is given as (c.f. (16)):

$$X_{3,\text{actual}}^{\ell+1} := \gamma\left(X_3^\ell, \phi\left(X_3^\ell, X_1^\ell, w_{31}\right) \coprod \phi\left(X_3^\ell, X_2^\ell, w_{32}\right)\right) \tag{17}$$

Since there is no dependence on S_{high} in equation (17) – which defines $X_{3,\text{actual}}^{\ell+1}$ – the desired propagation scheme can not arise, unless it is paradoxically already present at all scales S_{high}. If it is present at all scales, there is however only propagation along edges in $\underline{G}$, even if $S_{\text{high}} \approx S_{\text{reg.}}$, which would imply that the message passing network would not respect the graph structure of G. Hence $X_{3,\text{actual}}^{\ell+1} \nrightarrow X_{3,\text{desired}}^{\ell+1}$ does not converge as S_{high} increases.

B.2.2 Limit Propagation Schemes

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

Generic message functions: We first consider the possibility that the message function ϕ in (17) is implemented via an MLP using ReLU-activations: Assuming (for simplicity in notation) a one-hidden-layer MLP mapping features $X_i^\ell \in \mathbb{R}^{F_\ell}$ to features $X_i^{\ell+1} \in \mathbb{R}^{F_{\ell+1}}$ we have

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

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

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

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

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

$$\lim_{w_{ij} \to \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = 0 \tag{18}$$

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

$$\lim_{w_{ij} \to \infty} \phi(X_i^\ell, X_j^\ell, w_{ij})_a = \infty \tag{19}$$

For either choice of aggregation function $\coprod$ in (16) among "max", "sum" or "mean" the behaviour in (19) 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 (19) and continuity of ϕ we can conclude that components $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ of messages propagated along $\mathcal{E}_{\text{high}}$ for which $\left(W_3^\ell\right)_a > 0$ dominate over messages propagated along edges in $\mathcal{E}_{\text{reg.}}$. By (18), the former clearly also dominate over components $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ of messages propagated along $\mathcal{E}_{\text{high}}$ for which $\left(W_3^\ell\right)_a < 0$. This behaviour is irrespective of whether "max", "sum" or "mean" aggregations are employed. Hence the limit propagation scheme essentially only takes into account message channels $\phi(X_i^\ell, X_j^\ell, w_{ij})_a$ for which $(ij) \in \mathcal{E}_{\text{high}}$ and $\left(W_3^\ell\right)_a > 0$.

Similar considerations apply, if non-linearities are chosen as leaky ReLU. If instead of ReLU activations a sigmoid-nonlinearity σ like $\tanh$ is employed, messages propagated along $\mathcal{E}_{\text{large}}$ become increasingly uninformative, since they are progressively more independent of features X_i^ℓ and weights w_{ij}. Indeed, for sigmoid activations, the limits (18) and (19) are given as follows:

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

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

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

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

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

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

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

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

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_j^\ell \,\|\, w_{ij}\right]\right)\right)}{\sum\limits_{k\in\mathcal{N}(i)} \exp\left(\text{LeakyRelu}\left(\vec{a}^\top\left[WX_i^\ell \,\|\, WX_k^\ell \,\|\, w_{ik}\right]\right)\right)}, \tag{20}$$

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

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

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

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

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

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

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

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

since the denominator in (20) diverges. Hence in this case, propagation along $\mathcal{E}_{\text{reg.}}$ is essentially suppressed and features are effectively only propagated along $\mathcal{E}_{\text{high}}$.

C Coarse-graining Graphs and proof of (1)

In this Appendix – using the notation of Appendix B – we illustrate:

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

Using Theorem C.5, then yields the prove of the desired estimate

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The proof below then follows (Koke, 2025). An initial and more preliminary consideration of the problem was conducted in (Koke & Kutyniok, 2022; Koke, 2023). Further information may also be found in (Koke et al., 2023; 2024). We find:

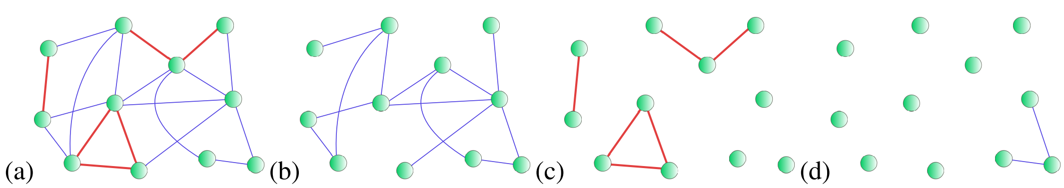

Figure 12: (a) Graph G with $\mathcal{E}_{\text{reg.}}$ (blue) & $\mathcal{E}_{\text{high}}$ (red); (b) $G_{\text{reg.}}$; (c) G_{high}; (d) $G_{\text{reg., exclusive}}$

Theorem C.3. We have

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

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

We here restate the proof for convenience. We use the notation $\Delta = L$.

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

$$R_z(\Delta) = (\Delta - zId)^{-1},$$
$$R_z(\Delta_{high}) = (\Delta_{high} - zId)^{-1}$$
$$R_z(\Delta_{reg.}) = (\Delta_{reg.} - zId)^{-1}$$

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

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

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

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

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

In our case, this translates to

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

or equivalently

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

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

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

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

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

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

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

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

which is precisely to say that

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

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

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

Our next step is to establish that

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

To this end we first note that the relation

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

provided to us by the second resolvent formula, implies

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

Thus we have

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

With this, we have

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

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

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

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

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

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

For S_{high} sufficiently large, we have

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

so that we may estimate

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

Thus we have now established

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

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

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

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

$$J^{\uparrow} \cdot J^{\downarrow} = P_0^{high} \tag{21}$$

and

$$J^{\downarrow} \cdot J^{\uparrow} = Id_{\underline{G}}. \tag{22}$$

Indeed,the relation (21) follows from the fact that the eigenspace corresponding to the eignvalue zero is spanned by the vectors $\{\mathbb{1}_R\}_R$, with $\{R\}$ the connected components of G_{high}. Equation (22) follows from the fact that

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

With this we have

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

To proceed, set

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

and

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

Then

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

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

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

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

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

Thus we have

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

This – in turn – implies

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

Using

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

we then have

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

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

As a corollary, we find

Corollary C.4. *We have*

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

Proof. This follows directly from the fact that

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

□

To prove (1), we establish the following theorem:

Theorem C.5. Consider a graph sequence G_n with $\|(L_n + \lambda Id)^{-1} - \tilde{J}_n(\tilde{L} + \lambda Id)^{-1}J_n\| \to 0$. Then we have $\|\psi(L_n) - \tilde{J}_n\psi(\tilde{L})J_n\| \to 0$ if ψ is complex differentiable and $\lim_{r\to\infty}\psi(r) = 0$.

Proof. We make use of the holomorphic functional calculus (c.f. e.g. (Koke & Cremers, 2024)) to establish

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

Since $\|(L_n + \lambda Id)^{-1} - \tilde{J}_n(\tilde{L} + \lambda Id)^{-1}J_n\| \to 0$ implies $\|(L_n - zId)^{-1} - \tilde{J}_n(\tilde{L} - zId)^{-1}J_n\| \to 0$ uniformly (in z) on compact sets (c.f. e.g. Arendt (2001)), we can apply dominated convergence, if we find an majorizing function that is integrable on Γ. But this is ensured by the decay of ψ. □

Choosing the function ψ to be given as $\psi(z) = e^{-tz}$ then establishes (1).

D GLOBAL LAPLACIAN PROPAGATION MATRICES, GENERALIZED FUNCTIONS, MEASURES AND ALL THAT

In this section we discuss global Laplacian propagation matrices, generalized functions and measures

D.1 COMPLEX MEASURES ON $\mathbb{R}_{\geqslant 0}$ AND THEIR THEORY OF INTEGRATION

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

In mathematics, a measure is a formal generalization of concepts such as length, area and volume.

More specifically, we are here interested in assigning a generalized notion of length (or mass) to subsets of the real half-line

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

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

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

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

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

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

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

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

Let us provide some examples:

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

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

Hence we may think of **measures as generalizations of functions**.

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

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

as

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

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

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

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

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

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

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

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

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

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

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

D.2 LAPLACE TRANSFORMS

We say a complex valued measure μ is finite if we have

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

for which

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

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

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

D.3 Global Laplacian Propagation matrices

A Global Laplacian Propagation matrix is then constructed by applying a function ψ arising as a Laplace transform to a graph Laplacian L. The resulting filter matrix $\psi(L) \in \mathbb{R}^{N\times N}$ acts on scalar graph signals $x \in \mathbb{R}^N$ via matrix multiplication; sending x to $\psi(L)\cdot x$:

$$x \mapsto \psi(L)\cdot x$$

E Proofs related to Generalization Ability

E.1 Generalization Ability of global Laplacian Propagation Matrices

In this section, we establish the generalization ability of global Laplacian propagation matrices as defined in Section 5.

Theorem E.1. We have that $\|\psi(L) - J^{\uparrow}\psi(\underline{L})J^{\downarrow}\| \leqslant \int_0^\infty |\hat{\psi}(t)|\eta(t)dt$ holds true.

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

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

In the notation of Section 5, we have $d|\mu|_{\hat{\psi}}(t) = |\hat{\psi}(t)|dt$ and hence

$$\begin{aligned}\|\psi(L) - J^{\downarrow}\psi(\underline{L})J^{\downarrow}\| &= \left\|\int_{\mathbb{R}_{\geqslant 0}} \left[e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\right] d\mu_{\hat{\psi}}\right\| \\ &\leqslant \int_{\mathbb{R}_{\geqslant 0}} \left\|e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\right\| |\hat{\psi}(t)|dt.\end{aligned}$$

□

Thus if $\eta(t) \equiv \left\|e^{-tL} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\right\| \approx 0$ on the support of $\hat{\psi}$, we also have $\|\psi(L) - J^{\uparrow}\psi(\underline{L})J^{\downarrow}\| \approx 0$. In this case, propagation as implemented via $\psi(L)$ is essentially the same as propagation via $J^{\downarrow}\psi(\underline{L})J^{\downarrow}$.

F Generalization and stability when $\|L - \tilde{L}\| \ll 1$

In this section we prove in addition to results in the main body of the paper also stability and generalization ability in the setting where for the Laplacians $L, \tilde{L}$ of two graphs $G, \tilde{G}$ defined on a common node set we have $\|L - \tilde{L}\| \ll 1$ (as opposed to the setting where one graph is a coarser version of another). We denote the collection of weight matrices by $\mathcal{W}$, the collection of biases by $\mathcal{B}$ and the (collection of) utilized global Laplacian propagation matrices used in the update rule "$X \mapsto \sum_k \psi_k(L)XW_k$" as Ψ. We denote the network by $\Phi_{\mathcal{W},\mathcal{B},\Psi}$ and write the generated embeddings for the node feature matrix X as $\Phi_{\mathcal{W},\mathcal{B},\Psi}(X)$. With this, we have:

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

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

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

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

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

We have

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

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

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

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

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

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

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

This proves the claim. □

F.1 Proof of Theorem5.2

The result in Theorem 5.2 is concerned with the graph-level setting; i.e. the setting where entire graphs are embedded into latent spaces. Before proving this result, we first prove a corresponding result for the node-level, where individual nodes in a graph are embedded. We will then use this node-level result (Thoerem F.2 below) to prove the graph-level Theorem 5.2.

In the node-level setting, we start by considering initial node-features X on G. We then fix a graph neural network Φ based on global Laplacian propagation schemes and consider two ways of generating embeddings on the graph G: On the one hand, we may simply generate embeddings with the network Φ on G. On the other hand, we may also project the node feature matrix X to $\underline{G}$ via $J^{\downarrow}$, apply ne the network Φ to the matrix $J^{\downarrow}X$ on $\underline{G}$ and then finally interpolate the generated node embeddings back to G via $J^{\uparrow}$.

The following result bounds the difference between these two respective node embeddings generated on the same graph.

Theorem F.2. Let $\Phi_{\mathscr{W},\mathscr{B},\Psi}$ be a K-layer deep Global-Laplacian-Propagation-based network. Assume $\sum_{i\in I}\|W_i^\ell\| \leqslant W$ and bound bias matrices in layer ℓ as $\|B^\ell\| \leqslant B$. Choose $C \geqslant \|\Psi_i(L)\|$ ($i \in I$) and w.l.o.g. assume $CW > 1$ (which can always be satisfied by choosing C large enough). Assume $\rho(J^{\uparrow}X) = J^{\uparrow}\rho(X)$ and if biases are enabled, assume $J^{\uparrow}\mathbb{1}_{\underline{G}} = \mathbb{1}_G$. Set $\max_{i\in I}\{\|\psi_i(L) - J^{\uparrow}\psi_i(\underline{L})J^{\downarrow}\|\} = \delta_1$ and define $\delta_2 = \max_{i\in I}\{\|\psi_i(L^{\uparrow})[J^{\downarrow}J^{\uparrow} - Id_{\underline{G}}]\|\}$. With this, we have that

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

It should be noted that the result above is more general than the setting considered in Section 5. In the setting considered in Section 5 we have $J^{\downarrow}J^{\uparrow} = Id_{\underline{G}}$ (in addition to $\rho(J^{\uparrow}X) = J^{\uparrow}\rho(X)$). There we thus automatically have $\delta_2 = 0$.

Proof. Let us define

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

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

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

and

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

We then have

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

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

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

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

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

From this, we find (assuming $\|J^{\uparrow}\|, \|J^{\downarrow}\| \leqslant 1$ for notational simplicity (and which is true in the setting of Section 5)), that

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

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

□

Let us move from the node-level to the graph-level. We first specify how graph-level latent embeddings arise:

Definition F.3. We aggregate embeddings $X \in \mathbb{R}^{N\times F}$ of individual nodes to graph-embeddings $\Omega(X) \in \mathbb{R}^F$ as $\Omega(X)_j = \sum_{i=1}^N |X_{ij}|\cdot\mu_i$. Here $\{\mu_i\}_i$ is the set of node-weights.

In a social network, a node weight $\mu_i = 1$ might e.g. signify that node i represents a single user. A weight $\mu_j > 1$ would indicate that node j represents a group of users.
Given such an aggregation of node embeddings into latent-embeddings of entire graphs, we may then relegate graph-level transferability back to node-level transferability:

Theorem F.4. Assuming $\Omega(\underline{X}) = \Omega(J^{\uparrow}\underline{X})$, we have in the setting of Theorem F.2 that $\|\Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Omega \circ \Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X)\| \leqslant \|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - J^{\uparrow}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X)\|$.

Proof. We note

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

To prove the claim from here, we only have to note that the aggregation method Ω as defined in Definition F.4 above is 1-Lipschitz (as a consequence of the reverse triangle inequality). The proof for the bidirectional setting proceeds analogously. □

This result then proves Theorem 5.2. Indeed: In the notation of Section 5, we have $F_{\omega} = \Omega(\Phi_{\mathscr{W},\mathscr{B},\Psi}(L_{\omega}, X))$ and $\underline{F} = \Omega(\Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X))$ Thus we have

$$
\|F_{\omega} - \underline{F}\| = \|\Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(L, X) - \Omega\circ\Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X)\| \leqslant \|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L_{\omega}, X) - J^{\uparrow}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X)\|.
$$

By Theorem F.2 and the fact that $[Id_{\underline{G}} - J^{\uparrow}J^{\downarrow}] = 0$, we have

$$
\|\Phi_{\mathscr{W},\mathscr{B},\Psi}(L_{\omega}, X) - J^{\uparrow}\Phi_{\mathscr{W},\mathscr{B},\Psi}(\underline{L}, J^{\downarrow}X)\| \lesssim \max_{k}\{\|\psi_k(L_{\omega}) - J^{\uparrow}\psi_k(\underline{L})J^{\downarrow}\|\},
$$

with "$\lesssim$" as per usual "denoting smaller than, up to a positive multiplicative constant".

Finally Theorem E.1 implies

$$
\|\psi_k(L_{\omega}) - J^{\uparrow}\psi_k(\underline{L})J^{\downarrow}\| \leqslant \int_0^{\infty} |\hat{\psi}_k(t)|\eta(t)dt = \int_{\mathbb{R}_{\geqslant 0}} \left\|e^{-tL_{\omega}} - J^{\uparrow}e^{-t\underline{L}}J^{\downarrow}\right\| |\hat{\psi}_k(t)|dt.
$$

Thus upon combining these steps, Theorem 5.2 is indeed proved.

G Additional Experimental Considerations

G.1 Additional details on Coarse Graining Example

Collapsing strongly connected clusters: Intuition and exact Definitions

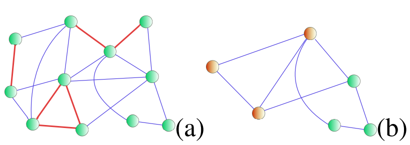

Figure 13: (a) G (stongly connected) clusters in red (b) Coarse grained $\underline{G}$

From a diffusion perspective, information in a graph equalizes faster along edges with large weights. In the limit where edge-weights within certain sub-graphs tend to infinity, information within these clusters equalizes immediately and such sub-graphs thus effectively behave as single nodes. We might thus consider a coarse grained graph $\underline{G}$ where these strongly connected clusters are indeed fused together and represented only via single nodes. The corresponding node set $\underline{\mathcal{G}}$ of $\underline{G}$ is then given by the set of connected components in G_{cluster} (c.f. Fig 14). 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 $J^{\uparrow}u = \sum_{R\in\underline{\mathcal{G}}} u_R \cdot \mathbb{1}_R$.

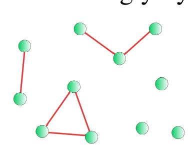

Figure 14: G_{cluster}

In this setting, we have (c.f. Appendix C) that

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

Here $w_{\text{high}}^{\min} \gg 1$ denotes the minimal edge weight inside the strongly connected clusters in G.

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

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

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

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

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

$$C_{ij}^{\text{Clmb}} = \frac{Z_i Z_j}{|R_i - R_j|} \tag{23}$$

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 (23).

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 (23) are modified similarly to generate the weight matrix $\underline{W}$.

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

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

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

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

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

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

The feature matrix

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

is a single row-vector given as

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

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

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

$$F_1 = F_2 = 64.$$

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

For the **resolvent** based global Laplacian propagation architecture, we set $\lambda = 1$ and and build filters using the $k = 1$ and $= 2$ matrices in $\Psi^{\text{Res}} = \{(z + \lambda)^{-k}\}_{k \in \mathbb{N}}$.

For the**based global Laplacian propagation architecture,** based global Laplacian propagation architecture, we set $t_0 = 1$ and and build filters using the $k = 1$ and $= 2$ matrices in $\Psi^{\text{Exp}} = \{e^{-(kt_0)z}\}_{k \in \mathbb{N}}$.

As aggregation, we employ the graph level feature aggregation scheme introduced in Definition F.3 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 Transferability and Generalization on Graphs generated via Stochastic Block Models

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

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

Experiment: Varying the Connectivity within the Communities: As discussed in detail in Appendix C, we desire that networks assign similar feature vectors to graphs with strongly connected communities and coarse-grained versions of these graphs, where these communities are collapsed to aggregate nodes. The higher the connectivity within these communities, the more similar should the feature vector of the original graph G and its coarsified version $\underline{G}$ be, as Appendix C established. In order to verify this experimentally, 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 15 below, we then plot the difference of feature vectors generated by **resolvent** and **exponential** global Laplacian propagation based models as well as GCN for G and $\underline{G}$ respectively. For each $p_{\text{connect}} \in [0, 1]$, results are averaged over 100 graphs randomly drawn from the same stochastic block model.

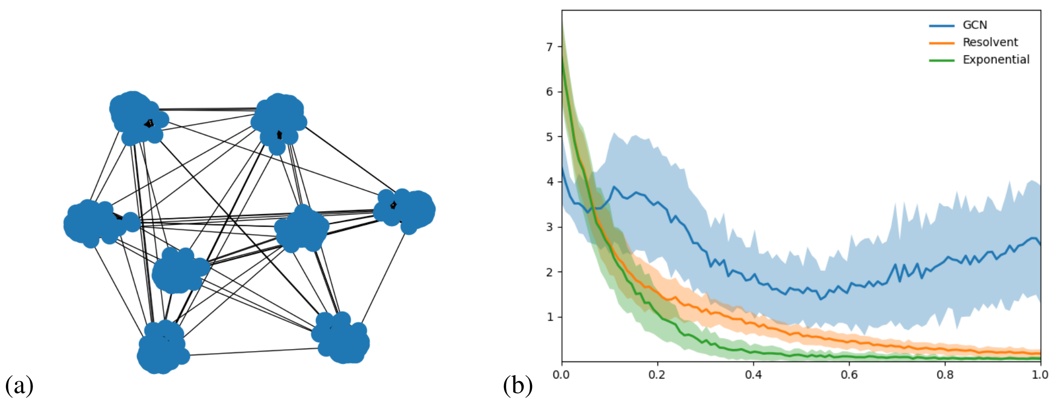

Figure 15: (a) Example Graph (b) Varying the parameter $p_{\text{connect}} \in [0, 1]$ for fixed $c_{\text{size}} = 20$, $p_{\text{inter}} = 2/c_{\text{size}}^2$ and $c_{\text{number}} = 10$.

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; any arbitrary choice of p_{inter} ensures a decay behavior as in Figure 15 for networks based on global Laplacian propagation matrices. A corresponding ablation study is provided below.

As can be inferred from Fig. 15, exponential- and resolvent based global Laplacian propagation methods produce more and more similar feature-vectors for G and its coarse-grained version $\underline{G}$, as the connectivity within the clusters is increased. As a reference, we plot GCN for which such a transferability result clearly does not hold.

G.3 Node Level Generalization and Graphs with varying Connectivity

We next consider popular citation networks (c.f. Appendix G.3 where each node corresponds to a piece of scientific writing. Labels correspond to the academic discipline of the paper and an edge implies a citation. We then expand individual nodes into connected k-cliques (c.f. Fig. 16). We might interpret this as further dissecting each article into subsections, which reference each other.

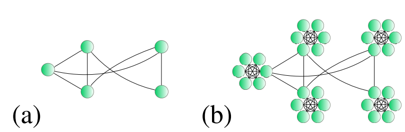

Figure 16: Individual nodes (a) replaced by k-cliques (b).

Both typical models (c.f. Appendix G.3) and global Laplacian propagation based methods were then trained on the same (k-fold expanded) train-set and asked to classify nodes in the (k-fold expanded) test-partition. The classification accuracy of methods not employing Laplace Transform filters decreases significantly with increasing clique size (c.f. Fig. 17). We can understand the underlying reason for this using GCN as an Example (c.f. again Appendix B for other methods):

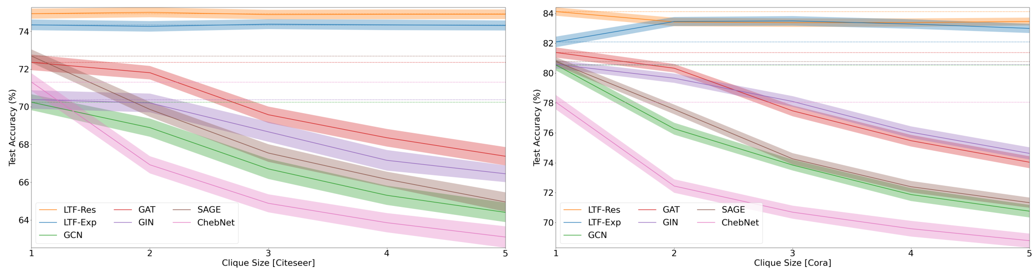

Figure 17: Node-Classification-Accuracy (↑) and uncertainty (for 100 runs) vs. clique size.

Inside a GCN-layer, a node feature matrix X is updated as $X \mapsto \hat{A}XW$, with the renormalized adjacency matrix $\hat{A}$ given as $\hat{A}_{ij} \sim A_{ij}/\sqrt{d_i d_j}$. As the degree d_i of each node increases (linearly) with increasing clique-size k, the message-strength $\hat{A}_{ij}$ between the respective cliques decreases as $\hat{A}_{ij} \sim 1/k$. Hence information propagation between the cliques becomes disrupted as k increases: GCN is more and more transferable between the given graph and a modified version where edges *between* cliques are removed. Models employing a global Laplacian propagation scheme are not afflicted by this shortcoming.

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

We train all models on a fixed learning rate of $\text{lr} = 0.1$. 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 over $\lambda \in \{0.0001, 0.0005\}$. Where applicable (e.g. not for He et al. (2021)) we choose a two-layer deep convolutional architecture with the dimensions of hidden features optimized over

$$K_\ell \in \{32, 64, 128\}. \tag{24}$$

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

For the LTF-models, we optimize depth over $K = 1, 2$ with hidden feature dimension optimized over the values in (24) as for baselines. We empirically observed in the setting of *unweighted* graphs, that rescaling the Laplacian as

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

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

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

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

The value λ for the resolvent is selected among

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

G.4 Transferability between Graphs discretizing a common Ambient Space

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 generalization ability.

G.4.1 Main Results

We consider the setting of two graphs G_1, G_2 discretely approximating the same ambient space (c.f. e.g. Fig. 1). This can be made mathematically precise using the concept of generalized norm resolvent convergence (c.f. e.g. (Post, 2012) for a discussion). Here we note the following: Given projection operators $J_i^{\downarrow}$ mapping from $\mathcal{M}$ to G_i and interpolation operators $J_i^{\uparrow}$ mapping from G_i to $\mathcal{M}$, we may measure the difference $\|e^{-t\Delta_{\mathcal{M}}} - J_i^{\uparrow} e^{-tL_i} J_i^{\downarrow}\| \leqslant \delta_i$ in diffusion flows on the respective spaces. The fidelity of the discrete approximation is then essentially determined by the size of $\delta_i \ll 1$. As discussed in detail in Appendix G.4.2, we have in this setting:

$$\|e^{-tL_1} - (J_1^{\downarrow} J_2^{\uparrow}) e^{-tL_2} (J_2^{\downarrow} J_1^{\uparrow})\| \lesssim (\delta_1 + \delta_2) \tag{25}$$

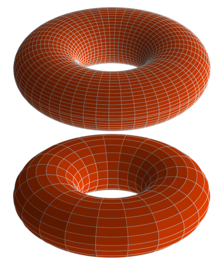

Figure 18: Torus Discretizations

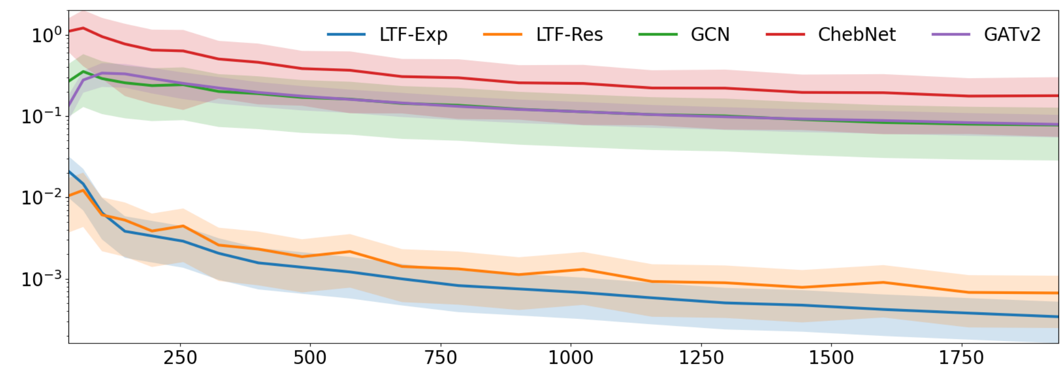

Figure 19: Transferability error $E = \|\Phi_1(J_1^{\downarrow} f) - (J_1^{\downarrow} J_2^{\uparrow})\Phi_2(J_2^{\downarrow} f)\|$ vs. # Nodes $N = |G_2| = 4|G_1|$

As an Example, we prove in Appendix G.4.2 below, that for the regular grid discretisation of the Torus and judiciously chosen translation operators $J_i^{\uparrow} J_i^{\downarrow}$, we have $\|e^{-t\Delta_{\mathcal{M}}} - J_i^{\uparrow} e^{-tL_i} J_i^{\downarrow}\||_{t>0} \leqslant \delta_i \to 0$ as the number of nodes in the approximating graphs G_i is increased. Given a fixed input signal $f \in L^2(\mathcal{M})$ on the Torus $\mathcal{M}$, eq. (25) together with Theorem F.2 then implies that thus also the generalization error $E = \|\Phi_1(J_1^{\downarrow} f) - (J_1^{\downarrow} J_2^{\uparrow})\Phi_2(J_2^{\downarrow} f)\|$ tends to zero as N increases. This error E measures the difference between sampling the signal f on $\mathcal{M}$ to G_1 and passing it through a GNN there, versus sampling f to G_2, applying the GNN on G_2 instead and subsequently transfering the output to G_1.

To numerically verify, that this generalization error indeed tends to zero for global Laplacian propagation based methods, we fix the number of nodes as $N = |G_2| = 4|G_1|$ in the respective graphs. We then plot E as a function of the number of nodes N for randomly initialized networks, with uncertainty calculated over 100 initializations.

We make use of the operators $J_i^{\uparrow\downarrow}$ defined in Appendix G.4.2. The function $f \in L^2(\mathcal{M})$ on the torus is chosen as

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

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

As evident from Fig. 1, the generalization error for global Laplacian propagation based methods tends to zero as N is increased. Additionally generalization errors of global Laplacian propagation based methods are consistently two orders of magnitude smaller than those of other networks.

G.4.2 THEORETICAL DETAILS

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

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

We will assume $J_i^{\downarrow}J_i^{\uparrow} = Id_{G_i}$, which is a justified assumption, as Example G.1 below elucidates. In this setting, we then have

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

We note

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

We consider:

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

Hence we have indeed established

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

Next let us consider an explicit example.

Example G.1. To this end, let us revisit the torus-setting introduced in Fig. 1.

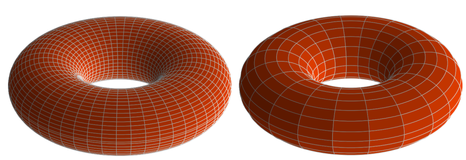

Figure 20: Distinct Torus Discretizations

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

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

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

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

A set of corresponding normalized eigenfunctions are given as

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

with corresponding eigenvalues

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

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

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

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

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

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

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

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

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

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

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

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

$$\|e^{-t\Delta_{\mathbb{T}}} - J^{\uparrow}e^{-t\Delta_N}J^{\downarrow}\| \to 0 \tag{26}$$

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

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

From this we observe

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

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

For the first summand, we already have

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

Hence let us investigate the second summand. We note

$$\begin{aligned} &\left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2} t} P_{k_1,k_2} - e^{-\lambda^N_{k_1,k_2} t} P^N_{k_1,k_2} \right) \right\| \qquad (27) \\ \leqslant &\left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} \left(e^{-\lambda_{k_1,k_2} t} - e^{-\lambda^N_{k_1,k_2} t} \right) P^N_{k_1,k_2} \right\| + \left\| \sum_{-\frac{N-1}{2} \leqslant k_1, k_2 \leqslant \frac{N-1}{2}} e^{-\lambda_{k_1,k_2} t} (P_{k_1,k_2} - P^N_{k_1,k_2}) \right\| \end{aligned}$$

For the first summand we note

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

We note

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

Using

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

we note

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

Hence it remains to bound the second summand in (27). We note

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

Next we note

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

It is not hard to see that

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

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

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

Where the lass claim follows from summability in k_1, k_2. Thus we have in total indeed established that (26) holds.