An Experimental Study of the Transferability of Spectral Graph Networks

Axel Nilsson and Xavier Bresson
School of Computer Science and Engineering
Nanyang Technological University (NTU), Singapore
contact@xelnilsson.com, xbresson@ntu.edu.sg

Abstract

Spectral graph convolutional networks are generalizations of standard convolutional networks for graph-structured data using the Laplacian operator. A common misconception is the unstability of spectral filters, i.e. the impossibility to transfer spectral filters between graphs of variable size and topology. This misbelief has limited the development of spectral networks for multi-graph tasks in favor of spatial graph networks. However, recent works have proved the stability of spectral filters under graph perturbation. Our work complements and emphasizes further the high quality of spectral transferability by benchmarking spectral graph networks on tasks involving graphs of different size and connectivity. Numerical experiments exhibit favorable performance on graph regression, graph classification and node classification problems on two graph benchmarks. The implementation of our experiments are available on GitHub for reproducibility.

Keywords: Graph networks, spectral convolution, transferability, benchmarking.

Introduction

Graph neural networks (Scarselli et al. 2009) is the class of networks that process data on graphs. Convolutional neural networks (LeCun et al. 1998), which have shown great performances on a variety of tasks defined on Euclidean domains such as computer vision, have been extended to graphs with spectral theory (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016) and spatial template matching (Kipf and Welling 2017). Graph convolutional networks (GCNs) have showed significant performances in myriad of domains, among others the representation of social networks to describe communities (Kipf and Welling 2017), fake news detection (Monti et al. 2019), chemistry (Gilmer et al. 2017), knowledge graphs (Schlichtkrull et al. 2017), Hamilton, Ying, and Leskovec 2017, physics (Cranmer et al. 2019), and recommendation systems (Monti, Bronstein, and Bresson 2017; Ying et al. 2018).

In this paper, we focus on the transferability of spectral GCNs. Spectral GCNs define learnable filters as parametric functions of the graph Laplacian operator. Transferability is an essential property of learning systems, related to their generalization capability. It demonstrates the ability of the system to learn consistent and discriminative features that can be used to make prediction for graphs unseen by the model during training. (Levie, Isufi, and Kutyniok 2019; Levie, Bronstein, and Kutyniok 2019) disproved the idea that spectral filters cannot be transferred to different graphs. Precisely, the authors in (Levie, Isufi, and Kutyniok 2019) showed that given a graph filter \( g \) and a small perturbation \( E \) with \( \| E \| \leq 1 \) on the graph Laplacian \( \Delta \) then the filter on the perturbed graph is a small perturbation of the original filter:

\[
\| g(\Delta) - g(\Delta + E) \| = O(\| E \|).
\]

In other words, the perturbation of the filter is only bounded by the perturbation of the graph, and filters are thus stable. Combining stability with equivariance, graph spectral filters are proven to be transferable. Besides, in (Levie, Bronstein, and Kutyniok 2019), they showed the robustness of spectral filters w.r.t. small graph perturbations by assuming the graphs are discretized from the same “continuous” space. They established the transferability error of a given filter \( g \):

\[
\text{Err}(g) \leq \text{Err}(\Delta) + \text{Err}($\text{Consistency}$).
\]

The transferability error is thus bounded by the perturbation of the Laplacian \( \Delta \) and the consistency error that vanishes for large graphs. Spectral GCNs are thus robust if graphs are discretized from the same underlying space. Because of the misconception of failure of spectral filter transfer, the use of spectral GCNs in a multi-graph setting has been scarce in the literature. (Knyazev et al. 2018) applied spectral GCNs to a set of multiple molecular graphs and (Ktena et al. 2018) to brain connectivity networks.

The goal of this work is to validate the theoretical results on the transferability of spectral filters with a corpus of experimental evidence. Specifically, we will show that Cheb-Nets (Defferrard, Bresson, and Vandergheynst 2016) performs favorably for the fundamental tasks of graph classification, graph regression and node classification using two graph benchmarks. Another class of spectral networks is CayleyNets (Levie et al. 2018), which may also be investigated in a future work.
ChebNets
(Defferrard, Bresson, and Vandergheynst 2016)

These graph networks define smooth spectral filters \( g_\theta \) parametrized with Chebyshev polynomials \( T_i \) applied to the normalised Laplacian operator \( \Delta \):

\[
\Delta = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}},
\]

where \( A \) is the adjacency matrix and \( D \) is the degree matrix. The spectral filters \( g_\theta \) are defined as

\[
g_\theta(\Delta)h = \sum_{i=0}^{k} \theta_i T_i(\Delta)h,
\]

where \( k \) is the number of Chebyshev polynomials, \( \theta \) are the learnable parameters, \( h \) is a signal defined on the graph, and \( \Delta = 2\lambda_{\max}^{-1} \Delta - I_n \) is the Laplacian re-normalized such that its eigenvalues are in the interval \([-1, 1]\), where the Chebyshev polynomials are orthogonal. A great computational advantage of these parametric filters is to define them with a recursive equation:

\[
\begin{cases}
T_0 = h \\
T_1 = \Delta T_0 \\
T_{k\geq2} = 2\Delta T_{k-1} - T_{k-2}
\end{cases}
\]

which reduces the computational complexity to \( O(n) \) for sparse graphs.

Graph Benchmarks

Recent projects (Hu et al. 2020; Dwivedi et al. 2020) have proposed benchmarks with collections of datasets to evaluate and compare GNNs. They not only provide openly accessible and peer-reviewed datasets but also leaderboards, helping to track the performance of different GNN models. In this work, we will use datasets from the Open Graph Benchmark (OGB version 1.2.3) (Hu et al. 2020) and Benchmarking-GNNs (Dwivedi et al. 2020b). The selected datasets are composed of multiple graphs and cover a variety of tasks such as graph regression (ZINC, ogbg-Molpcba), graph classification (ogbg-Molhiv) and node classification (CLUSTER, PATTERN). A summary of these datasets and some of their properties can be found in Table 1. We notice that three of the five selected datasets are made of molecular graphs. This is simply because there exist only a few real-world datasets of multiple graphs with varying sizes. Nonetheless, they exhibit significant differences, in respect of task and size, making them complementary.

PATTERN & CLUSTER

PATTERN and CLUSTER are node classification datasets for graphs generated synthetically with stochastic block models (SBMs) (Abbe 2017), commonly used to model communities in social networks. Each node has an intra-probability of being connected to a node in the same community and an extra-probability of being connected to a node in other communities. For CLUSTER, the task is to identify communities in a semi-supervised setting. Each graph has six SBM communities of 5 to 35 nodes, and one node in each community is labeled at random. For PATTERN, the task is pattern matching, that is recognizing pre-defined subgraphs embedded in larger graphs. There are 100 randomly generated patterns of 20 nodes with random features. The train/test/val splitting is 10K/2K/2K respectively for PATTERN and 10K/1K/1K for CLUSTER. More details on the construction of these datasets can be found in the paper (Dwivedi et al. 2020b).

ZINC

ZINC is a graph regression dataset composed of molecular graphs. The task is to predict the constrained solubility of each molecule (Jin, Barzilay, and Jaakkola 2018), a continuous variable. The accuracy is determined by the mean average error of the \( L1 \) loss over the test set. Node and edge features are categorical and correspond to the type of atoms and bonds. The number of graphs in the training/validation/test sets are respectively 10K/1K/1K graphs.

OGBG-MOL

ogbg-Molhiv and ogbg-Molpcba are two molecular datasets from the OGB benchmark, used for regression tasks. ogbg-Molhiv is the smallest of the two datasets with 41,127 graphs. The task is to predict if a given molecule would inhibit the replication of the HIV, cast as a binary label. It is evaluated by a ROC-AUC performance metric. ogbg-Molpcba represents a more difficult task as the set is larger with 437,929 graphs, and there are 128 properties to regress for each graph. Besides, the class distribution is skewed with only 1.4% of positive data. The performance is evaluated with average precision (AP). Both datasets are split by scaffolding to split the graphs based on their structure. This contrasts with ZINC which is randomly split, making easier to generalize (Hu et al. 2020).

ChebNet Architectures

Although it is essential to compare the performance of GCNs to contextualize a new model, there is no perfect approach to compare two GCNs. Therefore, each benchmark has defined a set of rules to enable the comparison of models. In this work, we will follow the strategy of each benchmark to compare ChebNets with the following popular models: GCN (Kipf and Welling 2017), GraphSage (Hamilton, Ying, and Leskovec 2017), GAT (Velickovic et al. 2018) and GIN (Xu et al. 2019).

All experiments are run on a single GeForce RTX 2070 8GB GPU. The source code is available at: https://github.com/Axeln78/Transferability-of-spectral-gnns

Benchmarking-GNNs

In Benchmarking-GNNs (Dwivedi et al. 2020b), a budget of 100,000 learnable parameters with four convolutional layers are fixed to compare the performance with the same learning capacity. We will follow this constraint when designing ChebNets for CLUSTER, PATTERN and ZINC. The learning hyperparameters are the same as in (Dwivedi et al. 2020b).
Chebyshev order is $k$, a hidden dimensionality of 300 followed by three linear layers, five convolutional layers, a mean-pooling layer with normalisation and no dropout. Finally, the score is averaged over four runs with pre-selected seeds. 

For the selected OGB datasets, ogbg-Molhiv and ogbg-Molpcba, we will adhere to the default architectures described in (Hu et al. 2020) i.e. those used for GIN and GCN. They are composed of an Atom- and Edge- Embedding layer, five convolutional layers, a mean-pooling layer with a hidden dimensionality of 300 followed by three linear layers with a tuned dropout ratio $\in \{0.0, 0.5\}$. The learning hyperparameters are the default OGB values, with a learning rate of $10^{-3}$ for the Adam optimizer and a batch size of 128. The result is the average over ten runs with random seeds.

A summary of the ChebNet architectures for all datasets is available in Table 2. Finally, observe that the OGB leaderboard has no strict rules about model architecture, parameter budget, and hyperparameter selection. Therefore we will only provide the two models GIN and GCN, which have similar architectures as a basis of comparison, and discuss more informally models that are on the leaderboard as of this writing.

We also notice that both benchmarks do not have any spectral GCNs to compare to in their leaderboard.

### Values of $k$ and $\lambda_{\text{max}}$

We select the number of Chebyshev polynomials to be $k = 5$ for PATTERN and CLUSTER. This value is related to the size of the communities (5-35 nodes). For ZINC, the chosen Chebyshev order is $k = 2$ as the graphs are small (9 to 35 nodes). For ogbg-Molhiv and ogbg-Molpcba, the order of polynomial is selected to be $k = 3$.

For all ChebNets, the spectral parameter $\lambda_{\text{max}} = 2$ is fixed, as the largest eigenvalues can sometimes be numerical unstable to compute on certain graphs. Besides, fixing $\lambda_{\text{max}} = 2$ is mathematically justified as the spectrum of the normalized Laplacian is bounded by this value (Chung and Graham 1997). Another advantage is to reduce the computational cost and allow to compare with papers using this approximation (Knyazev et al. 2019, 2018).

### Numerical Results

Overall, numerical experiments show that ChebNets perform favorably well compared to other popular GCN models with comparable architectures. Table 3 reports the performance of ChebNets for each task.

#### CLUSTER

ChebNets outperform all models for the CLUSTER dataset and perform at the level of deeper models found in the leaderboard of the benchmark. In fact, the strong performance implies that ChebNets hold a good inductive bias for identifying community clusters in a semi-supervised node classification setting. This was expected as Laplacian-based clustering techniques have showed significant performance for community detection (Von Luxburg 2007). This is in contrast to the current literature as the a commonly used model for this task has been GCNs (Kipf and Welling 2017), which perform worse (25 percentage points). Furthermore, if we consider that this dataset is a set of discretizations of a continuous manifold, then this result is a salient confirmation of the theory of (Levie, Bronstein, and Kutyniok 2019).

#### PATTERN

ChebNets perform better than the other models for PATTERN and performs closely to GIN. Both models are have significantly higher accuracy than the rest. Although is expected that GIN performs well on tasks of pattern and graph matching, it is striking that the ChebNet performs equally well. Additionally, we observe no significant improvement in accuracy with deeper models in the benchmark leaderboard for this task.

#### ZINC

The ZINC experiment allows to compare ChebNets with a range of models with a similar budget of learnable parameters. Comparing to the leaderboard in (Dwivedi et al. 2020b), ChebNets not only performs significantly better than spatial GCN models, they exhibit superior performances over models with the same depth although some are using edge features like GAT or Gated GCN (Bresson and Laurent 2018).

#### OGB-MOL

For the two OGB tasks, ChebNets achieve better performance than the two models of comparison, GIN and GCN. Unexpectedly, ChebNets outperform GIN, a model designed...
Table 2: Summary of all model architectures used. E stands for Embedding, AE for AtomEmbedding, MP for Mean Pooling, L for Linear, ChN for ChebNets layer, and No-RC for no residual connections used. Each layer type is followed by a number indicating the output dimension.

| Dataset          | Model Architecture          | Hyperparam. |
|------------------|----------------------------|-------------|
| CLUSTER          | 7 -E70 -ChN70 -ChN70 -ChN70 -MP70 -L35 -L17 -L6 | k = 5       |
| PATTERN          | 3 -E70 -ChN70 -ChN70 -ChN70 -MP70 -L35 -L17 -L2 | k = 5       |
| Zinc             | 28 -E106 -ChN106 -ChN106 -ChN106 -MP106 -L53 -L26 -L1 (No-RC) | k = 2       |
| ogbg-Molhiv      | -AE300 -ChN300 -ChN300 -ChN300 -ChN300 -MP300 -L150 -L75 -L1 | k = 3       |
| ogbg-Molpcba     | -AE300 -ChN300 -ChN300 -ChN300 -ChN300 -MP300 -L150 -L75 -L128 | k = 3       |

Table 3: Numerical study of ChebNets on several datasets. The models are ranked w.r.t. performance. The proposed ChebNets are in bold. The performances of the models identified with * are taken from the leaderboards of different benchmarks as of Nov. 9th, 2020. The results are the average and standard deviation over four runs for tasks from Benchmarking-GNNs and ten for OGB. ChebNets show favorable performance compared to other models by achieving the best performance in all five tasks.

| Dataset          | Model     | # parameters | Accuracy     | Metric                  |
|------------------|-----------|--------------|--------------|-------------------------|
| CLUSTER          | ChebNet   | 102,535      | 73.13 ± 0.64 | Weighted accuracy in % (higher is better) |
|                  | GAT*      | 110,700      | 57.73 ± 0.32 |
|                  | GIN*      | 100,884      | 49.64 ± 2.09 |
|                  | GCN*      | 103,077      | 47.82 ± 4.91 |
|                  | GraphSage*| 99,139       | 44.89 ± 3.70 |
| PATTERN          | ChebNet   | 102,183      | 85.75 ± 0.02 |
|                  | GIN*      | 100,884      | 85.59 ± 0.01 |
|                  | GraphSage*| 98,607       | 78.20 ± 3.06 |
|                  | GAT*      | 109,936      | 75.82 ± 1.82 |
|                  | GCN*      | 100,923      | 74.36 ± 1.59 |
| Zinc             | ChebNet   | 101,230      | 0.360 ± 0.028| MSE (lower is better) |
|                  | GIN*      | 103,079      | 0.387 ± 0.015|
|                  | GCN*      | 103,077      | 0.459 ± 0.006|
|                  | GraphSage*| 94,977       | 0.468 ± 0.003|
|                  | GAT*      | 102,385      | 0.475 ± 0.007|
| ogbg-Molhiv      | ChebNet   | 1,465,351    | 0.7631 ± 0.0127| ROC-AUC (higher is better) |
|                  | GIN*      | 527,701      | 0.7606 ± 0.0097|
|                  | GCN*      | 1,885,206    | 0.7558 ± 0.0140|
| ogbg-Molpcba     | ChebNet   | 1,475,003    | 0.2306 ± 0.0016| Average precision (AP) (higher is better) |
|                  | GIN*      | 1,923,433    | 0.2266 ± 0.0028|
|                  | GCN*      | 565,928      | 0.2020 ± 0.0024|

Conclusion

This experimental work investigates the transferability capacity of spectral GCNs on two graph benchmarks. Numerical experiments demonstrate that ChebNets perform better than most popular spatial GCNs with comparable parameter budgets. These numerical experiments strongly support recent analytical results [Levie, Isufi, and Kutyniok 2019, Levy, Bronstein, and Kutyniok 2019] that spectral GCNs can compete at least as well as other spatial GCNs in the multi-graph setting. Such results are promising to encourage
the development of new spectral networks.

Acknowledgments
XB is supported by NRF Fellowship NRFF2017-10.

References
Abbe, E. 2017. Community detection and stochastic block models: recent developments. arXiv:1703.10146 [cs, math, stat].

Bresson, X.; and Laurent, T. 2018. Residual Gated Graph ConvNets. arXiv:1711.07553 [cs, stat].

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. arXiv:1312.6203 [cs].

Chung, F. R.; and Graham, F. C. 1997. Spectral graph theory. 92. American Mathematical Soc.

Cranmer, M. D.; Xu, R.; Battaglia, P.; and Ho, S. 2019. Learning Symbolic Physics with Graph Networks. arXiv:1909.05862 [astro-ph, physics:physics, stat].

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. arXiv:1606.09375 [cs, LG] 9.

Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; and Bresson, X. 2020a. Benchmarking Graph Neural Networks. arXiv:2003.09982 [cs, stat].

Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; and Bresson, X. 2020b. Benchmarking Graph Neural Networks. arXiv:2003.09982 [cs, stat].

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. arXiv:1704.0212 [cs].

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. arXiv:1706.02216 [cs, stat].

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. arXiv:2005.06687 [cs, stat].

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. arXiv preprint arXiv:1802.04364.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs, stat].

Knyazev, B.; Lin, X.; Amer, M. R.; and Taylor, G. W. 2018. Spectral Multigraph Networks for Discovering and Fusing Relationships in Molecules. arXiv:1811.09595 [cs, stat].

Knyazev, B.; Lin, X.; Amer, M. R.; and Taylor, G. W. 2019. Image Classification with Hierarchical Multigraph Networks. arXiv:1907.09000 [cs].

Ktena, S. I.; Parisot, S.; Ferrante, E.; Rajchl, M.; Lee, M.; Glocker, B.; and Rueckert, D. 2018. Metric learning with spectral graph convolutions on brain connectivity networks. NeuroImage 169: 431–442.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11): 2278–2324.

Levie, R.; Bronstein, M. M.; and Kutyobi, G. 2019. Transferability of Spectral Graph Convolutional Neural Networks. arXiv:1907.12972 [cs, stat].

Levie, R.; Isufi, E.; and Kutyobi, G. 2019. On the Transferability of Spectral Graph Filters. arXiv:1901.10524 [cs, stat].

Levie, R.; Monti, F.; Bresson, X.; and Bronstein, M. M. 2018. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing 67(1): 97–109.

Monti, F.; Bronstein, M.; and Bresson, X. 2017. Geometric Matrix Completion with Recurrent Multi-Graph Neural Networks. arXiv:1704.06803 [cs, LG] 11.

Monti, F.; Frasca, F.; Eynard, D.; Mannion, D.; and Bronstein, M. M. 2019. Fake News Detection on Social Media using Geometric Deep Learning. arXiv:1902.06673 [cs, stat].

Scarselli, F.; Gori, M.; Ah Chung Tsoi; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. IEEE Transactions on Neural Networks 20(1): 61–80. ISSN 1045-9227, 1941-0093. doi:10.1109/TNN.2008.2005605.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2017. Modeling Relational Data with Graph Convolutional Networks. arXiv:1703.06103 [cs, stat].

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903 [stat, ML] 12.

Von Luxburg, U. 2007. A tutorial on spectral clustering. Statistics and computing 17(4): 395–416.

Weisfeiler, B.; and Lehman, A. A. 1968. A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsia 2(9): 12–16.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? arXiv:1810.00826 [cs, stat].

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 974–983. doi: 10.1145/3219819.3219890.