A SPECTRAL PERSPECTIVE ON GCNS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we study the behavior of standard GCNs under spectral manipulations. We relate the numerical performances of a GCN through bandpass filtering: we empirically show that most of the necessary and used information for nodes classification is contained in the low-frequency domain, which contrasts with the standard use of high-frequencies in signal processing. In particular, it is possible to obtain accuracies at a state-of-the-art level with simple MLPs that rely only on a few low frequencies: this is surprising because contrary to GCNs, no cascade of filtering along the graph structure is involved. We also study a simple procedure to adjust the trade-off between smoothing and high-frequency propagation for deeper GCNs. We conduct our experiments on standard benchmarks.

1 INTRODUCTION

Graph Convolutional Networks (GCNs) are the state of the art in semi-supervised settings, such as community detection (Kipf & Welling, 2016). They correspond to Graph Neural Networks (GNNs) that propagate graph features through a cascade of linear operator and non-linearities, while exploiting the graph structure through a linear smoothing operator. However, the graph principles that allow GCNs to obtain good performances remain unclear. It is suggested in Li et al. (2018) that GCNs are eager to over-smooth their representation, which indicates they average too much neighborhood nodes and dilute classification information. The smoothing is generally interpreted as a low-pass filtering through the graph Laplacian, and finding a way to exploit high-frequencies of the graph Laplacian is an active research question (Oono & Suzuki, 2019). The results of our work actually suggest that graph Laplacian high-frequencies have actually a minor impact on the classification performances of a standard GCN, as opposed to standard Convolutional Neural Networks for vision.

Graph Signal Processing (GSP) is a popular field whose objective is to manipulate signals spectrum whose topology is given by a graph. Typically, this graph has a non-Euclidean structure, however many central theoretical results (Hammond et al., 2011) are based on an analogy with Euclidean, regular grids. For instance, a spectral component or frequency has to be understood as an eigenvalue of the Laplacian, yet it thus suffers from intrinsic issues such as isotropy (Oyallon, 2020). Those principles are very appealing because they allow to use the dense literature of harmonic analysis, on graphs. Thus, this literature is at the core of many intuitions and drives many key ingredients of a GCN design, which evokes standard tools of signal processing: convolutions, shift invariance, wavelets, Fourier (Bronstein et al., 2017), etc. Here, we certainly observe several limits of this analogy in the context of semi-supervised settings: for instance, we observe that discarding the high-frequencies has a minor impact on a GCN behavior.

Spectral clustering is a rather different point of view which studies node labelling in unsupervised contexts: it generally relies on generative models based on the graph spectrum. The main principle is to consider the eigenvectors corresponding to the smallest non-zero eigenvalues, referred as Fiedler vectors (Doshi & Eun, 2020): those directions allow to define clusters, depending on the sign of a feature. Several theoretical guarantees can be obtained in the context of Stochastic Block Model approximation (Rohe et al., 2011). Our paper proposes to establish a clear link with this approach: we show that the informative graph features are located in a low frequency band of the graph Laplacian and do not need extra graph processing tools to be efficiently used by a deep supervised classifier.

The major contribution of this paper is to show that a Multi Layer Perceptron (MLP) fed with node features and spectral coordinates of a given node is competitive with a GCN, whereas GCNs

typically heavily relies on the graph structure. Other contributions are as follow: **(a)** First we show that most of the necessary information exploited by a semi-supervised GCN is actually located in the very first eigenvectors of a Laplacian. **(b)** We numerically show that the high-frequency eigenvalues are less informative and that a trained GCN is stable to them. **(c)** We establish a clear link between discriminative power and low-frequencies in a supervised context, thanks to MLP classifiers that incorporate no other graph structures. **(d)** This MLP method allows to potentially deal with non-sparse adjacency matrix, based on a low rank factorization. To our knowledge, this is the first competitive results obtained with a MLP on transdusctive tasks. **(e)** We however show that manipulating the high-frequencies of a graph Laplacian (by reducing or on the contrary enhancing them) can lead to some minor performance boosts on GCNs.

We now discuss the organization of the paper: first, we discuss the related work in Sec. 2. We explain our notations as well as our framework in Sec. 3. Then, we study a low-rank approximation of a graph Laplacian in Sec. 4.1. Finally, Sec. 4.2 proposes several experiments to study the impact of high-frequencies on GCNs. A basic code is provided in the supplementary materials, and our code will be released on an online public repository at the time of publication, with the exact commands to reproduce our experiments.

2 Related Work

GCNs and Spectral GCNs Introduced in Kipf & Welling (2016), GCNs allow to deal with large graph structure in semi-supervised classification contexts. This type of model works at the node level, meaning that it uses locally the adjacency matrix. This approach has inspired a wide range of models, such as linear GCN (Wu et al., 2019), Graph Attention Networks (Veličković et al., 2017), GraphSAGE (Hamilton et al., 2017), etc. In general, this line of work does not consider directly the graph Laplacian. Another line of work corresponds to spectral methods, that employ filters which are designed from the spectrum of a graph Laplacian (Bruna et al., 2013). In general, those works make use of polynomials in the Laplacian (Defferrard et al., 2016), which are very similar to an anisotropic diffusion (Klicpera et al., 2019). All those references share the idea to manipulate bandpass filters that discriminate different ranges of frequencies.

Over-smoothing in GCNs In the context of GCN, Li et al. (2018) is one of the first paper to notice that cascading low-pass filters can lead to a substantial information loss. Zhao & Akoglu (2019); Yang et al. (2020) proposes to introduce regularizations which address the loss of information issues. Cai & Wang (2020); Oono & Suzuki (2019) study the spectrum of a graph Laplacian under various transform, yet they consider the spectrum globally and in asymptotic settings, with a deep cascade of layers. Huang et al. (2020); Rong et al. (2019b) introduce data augmentations, whose aim is to alleviate overs-moothing in deep networks: we study GCNs without this ad-hoc procedure.

Spectral clustering and low rank approximation As the litterature about spectral clustering is large, we mainly focus on the subset that connects directly with GCN. Mehta et al. (2019) proposes to learn an unsupervised auto-encoder in the framework of a Stochastic Block Model. Oono & Suzuki (2019) introduces the Erdös – Renyi model in the GCN analysis, but only in an asymptotic settings. Loukas & Vandergheynst (2018) studies the graph topology preservation under coarsening of the graph, which could be a potential direction for future works.

Node embedding As we will see below, our MLP approach can be understood as an embedding at the node level. For instance, Aubry et al. (2011) applies a spectral embedding combined with a diffusion process for shape analysis, which allows point-wise comparisons. We should also point Deutsch & Soatto (2020) that uses a node embedding, based on the spectrum of a quite modified graph Laplacian, obtained from on a measure of node centrality.

Graph Scattering Networks (GSN) This class model explicitly employs band-pass based on the spectrum of a graph Laplacian and it is thus necessary to review it. Gao et al. (2019); Gama et al. (2018; 2019) are a class of neural networks built upon an analogy with a Scattering Transform (Mallat, 2012). They typically rely on a cascade of wavelets followed by an absolute value: the objective of each wavelet is to separate multi-scale information into dyadic bandpass filters. This method relies heavily on each eigenvectors of the Laplacian and the dyadic space is typically constructed

from a diffusion process at dyadic intervals. Interferometric Graph Transform on the other hand relies on the concept of demodulation, which is clear in the context of Lie groups but unclear for semi-supervised tasks (Oyallon, 2020).

GCN stability Stability of GCNs has been theoretically studied in Gama et al. (2019), which shows that defining a generic notion of deformations is difficult. Surprisingly, it was noted in Oyallon (2020) that stability is not a key component to good performances. Stability of GCN has also been investigated in Verma & Zhang (2019) but only considers neural networks with a single layer and relies on the whole spectrum of the learned layer. Keriven et al. (2020) considers stability of GCNs, their results rely on an implicit Euclidean structure: it is unclear if this holds in our settings. Sun et al. (2020) is one of the first work to study adversarial examples linked to the node connectivity and introduces a loss to reduce their effects. Zhu et al. (2019) also addresses the stability issues by embedding GCNs in a continuous representation. None of these work directly related a trained GCN to spectral perturbations.

3 METHOD

Our initial graph data are node features $x \in \mathbb{R}^{K \times P}$ where K is the size of a graph with adjacency matrix $\mathcal{A} \in \mathbb{R}^{K^2}$ and P is the number of node-features. We consider GCNs as introduced in (Kipf & Welling, 2016), which correspond to GNNs that propagate features graph input $x_0 \triangleq x$ through a cascade of layers, via the iteration:

$$x_{n+1} \triangleq \sigma\Big(Ax_nW_n\Big)\,, \tag{1}$$

where A is a linear averaging obtained from the adjacency matrix $\mathcal{A}$, σ a point-wise non-linearity and W_n a parametrized affine operator. Note that if $A = \boldsymbol{I}I$, then Eq. 1 is simply an MLP. In the semi-supervised setting, a final layer x_N is fed to a supervised loss ℓ and $\{W_0, ..., W_{N-1}\}$ are trained in an end-to-end manner to adjust the label of each node. In this setting, we let $\mathcal{D}$ be the diagonal matrix of vertex degree, and we consider the first order GCN adjacency matrix as defined in Kipf & Welling (2016):

$$A \triangleq \frac{1}{2}\big(\boldsymbol{I} + \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\big)\,. \tag{2}$$

Here, we normalize A such that $\|A\| = 1$. We remark that other smoothings A have been proposed (using self-loops in the normalization), however, we tried several different possibilities that did not change our numerical conclusions, thus we decided to keep the simplest formulation to be handled mathematically. For numerical results, please refer to the supplemental materials. Note also that ignoring $\mathcal{D}$, $\boldsymbol{I} + \mathcal{A}$ is often interpreted as a low-pass filter, as for a given feature $x = (x_i)_i$, it explicitly performs a local averaging of a node i:

$$[(\boldsymbol{I} + \mathcal{A})x]_i = \eta x_i + \sum_{j \to i} \mathcal{A}_{i,j} x_j\,. \tag{3}$$

We are interested in analyzing the properties of spectral approximations of A. We consider the decreasing set of eigenvalues $\Lambda = \{\lambda_k\}_{k \geq 0}$ of A, such that $\lambda_k \geq \lambda_{k+1}$, and we denote by e_k the k-th eigenvector corresponding to $\lambda_k \in \Lambda$. We remind that $\Lambda \subset [0, 1]$ and that $\lambda_0 = 1$ can be interpreted as the lowest frequency of the graph Laplacian. Since the adjacency matrix is normalized, one basis of λ_0's eigenspace is constituted by the constant vectors of 1 supported on each connected component. We then note that:

$$A = \sum_{k \geq 0} \lambda_k e_k e_k^T\,. \tag{4}$$

In our work, we are interested in studying operators Π on A, of the type:

$$\Pi(A) = \sum_{k \geq 0} \Pi(\lambda_k) e_k e_k^T\,. \tag{5}$$

For instance, if $\Pi(\lambda) = 2(1 - \lambda)$, then $\Pi(A)$ is the normalized graph Laplacian. The next section studies empirically the property of such spectral transformations in the context of GCNs and simpler architectures.

4 NUMERICAL EXPERIMENTS

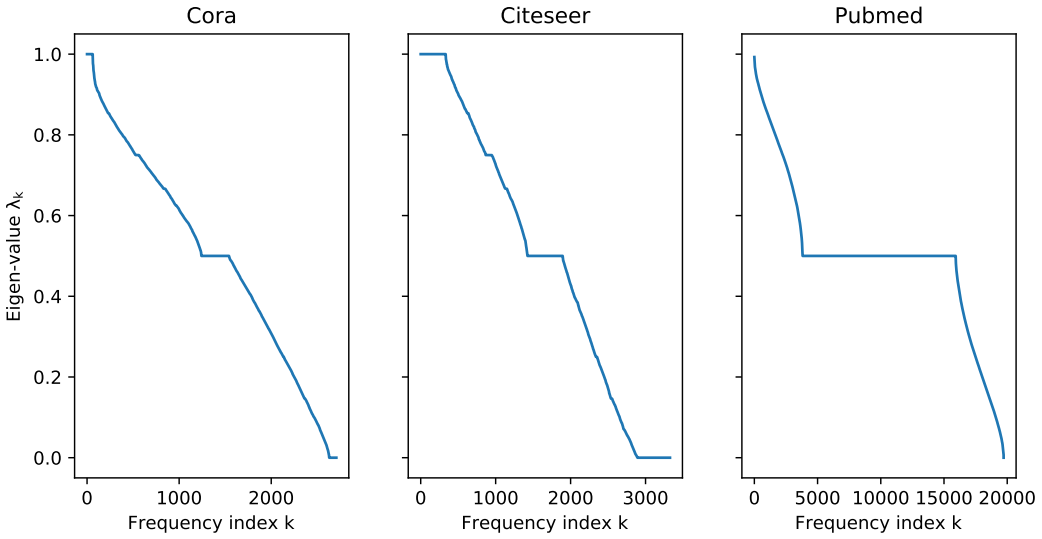

Figure 1: Spectrum Λ of A. Note that the eigenvalues are decreasing, displayed with multiplicity.

Matching previous work practice, we focus on three classical benchmark datasets varying in graph size and feature type: Cora, Citeseer and Pubmed (Sen et al., 2008). The task consists in classifying the research topic of papers in three citation datasets. Those tasks are transductive, meaning all node features are accessible during training. We apply the full-supervised training fashion used in Huang et al. (2016), Chen et al. (2018), and Rong et al. (2019b) on all datasets in our experiments. Fig. 1 plots the 3 spectra Λ in decreasing order for each of those datasets. Note that Pubmed has no connected component, and that the decay of its spectrum is fast compared to Cora or Citeseer, which indicates a low-dimensional structure (Belkin & Niyogi, 2002). Note that they are relatively sparse and not necessarily fully supported on their connected component. The statistics of all datasets are listed in the supplemental materials.

We follow the standard training procedure of Rong et al. (2019a). We choose σ to be the standard ReLU non-linearity (Krizhevsky et al., 2012). Unless specified otherwise, the weights of our models (either GCN or MLP) are optimized via Adam, with an initial learning rate 0.01 and weight decay of 0.001, during 800 epochs. We use by default a dropout of 0.5. Our model consists in GCN layers with 2 hidden layers of size 128. In all experiments, we cross validate our hyper-parameters on a validation set, using an early stopping at epoch 400. Unless specified otherwise, each plot is obtained by an average over at least 3 different seeds.

4.1 LOW RANK APPROXIMATION

Band-pass filters for GCNs Here, we consider the following projections:

$$\Pi_{\geq K-k}(\lambda) = \begin{cases} \lambda, & \text{if } \lambda \leq \lambda_{K-k} \\ 0, & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \Pi_{\leq k}(\lambda) = \begin{cases} \lambda, & \text{if } \lambda \geq \lambda_k \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

k is adjusted to retain only a portion of the spectrum. Here, $\Pi_{\geq K-k}$ and $\Pi_{\leq k}$ can be interpreted respectively as high-pass and low-pass filters. Fig. 2 and Fig. 3 report the respective numerical performance after applying those filters to the averaging A.

On Fig. 2, we observe that retaining only very few frequencies (less than 10%) does not degrade much the accuracy of the original network. This is well aligned with the intuition of Li et al. (2018) which studies empirically the over-smoothing phenomenon, as it indicates that the GCN uses mainly the low frequency domain. For Pubmed, using almost all the high-frequencies is required to recover the 5% accuracy necessary to get the original accuracy of the GCN. Interestingly, deeper GCNs

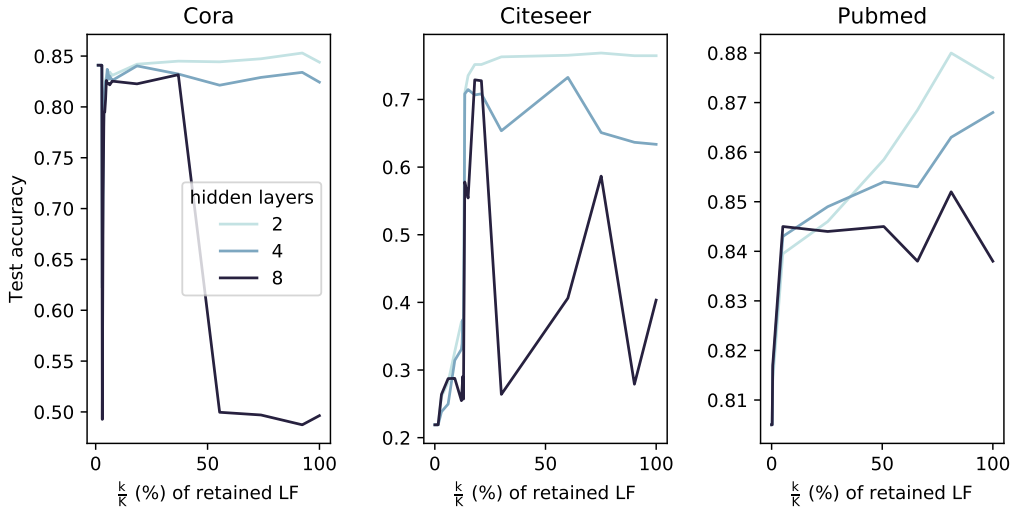

Figure 2: Test accuracies reached by a GCN as a function of the Low-Frequencies (LF) band $[\lambda_k, \lambda_0]$ retained by $\Pi_{\leq k}$, for various depths (100% corresponds to the full spectrum, including low frequencies).

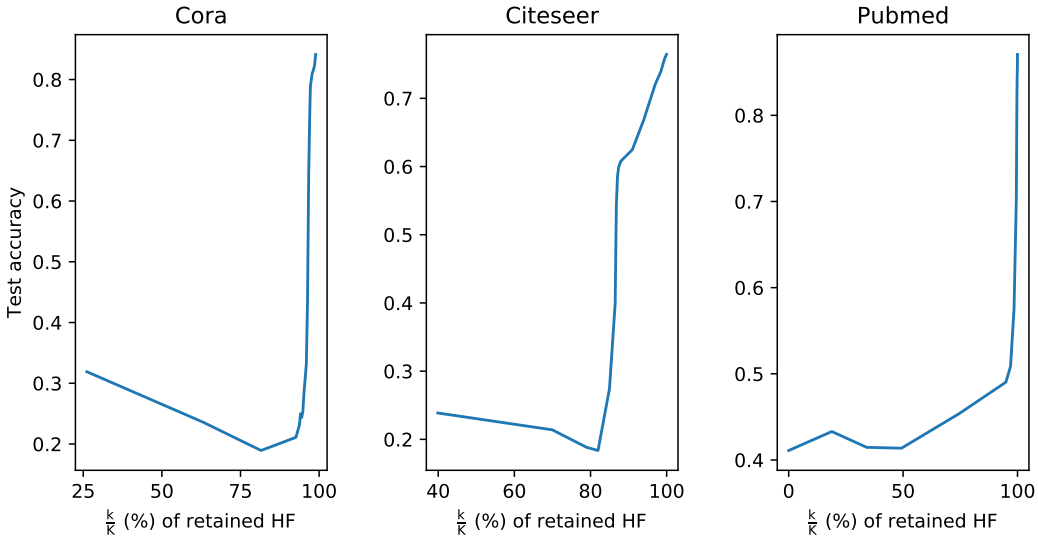

Figure 3: Test accuracies reached by a GCN as a function of the High-Frequencies (HF) band $[\lambda_K, \lambda_{K-k}]$ retained by $\Pi_{\geq K-k}$, for a GCN of depth 2 (100% corresponds to the full spectrum).

seem to benefit from the high frequency ablation, yet their accuracy remains below their shallow counter-part. Fig. 3 indicates that the necessary information for semi-supervised classification is contained in the low-frequency: dropping the latter leads to substantial accuracy drop, even for a shallow GCN.

MLP ablation We further study how informative low-frequencies are via a spare ablation experiment based on a MLP. Each dataset has a set of predefined features x, that we transform through the concatenation $\tilde{x}_k = [x, e_1^T, ..., e_k^T] \in \mathbb{R}^{K \times (P+k)}$. Informally, $\tilde{x}_k$ corresponds to the original features x augmented with the k first spectral components. For a fair evaluation, we used exactly the same hyper-parameters as for the experience above. Fig. 4 reports the accuracy of our MLP trained on $\tilde{x}_k$ as a function of k. We see that using a fraction $\frac{k}{K} \leq 20\%$ of the eigenvectors allows to recover the performance of a GCN trained end-to-end. More surprisingly, we note that as k increases, the accu-

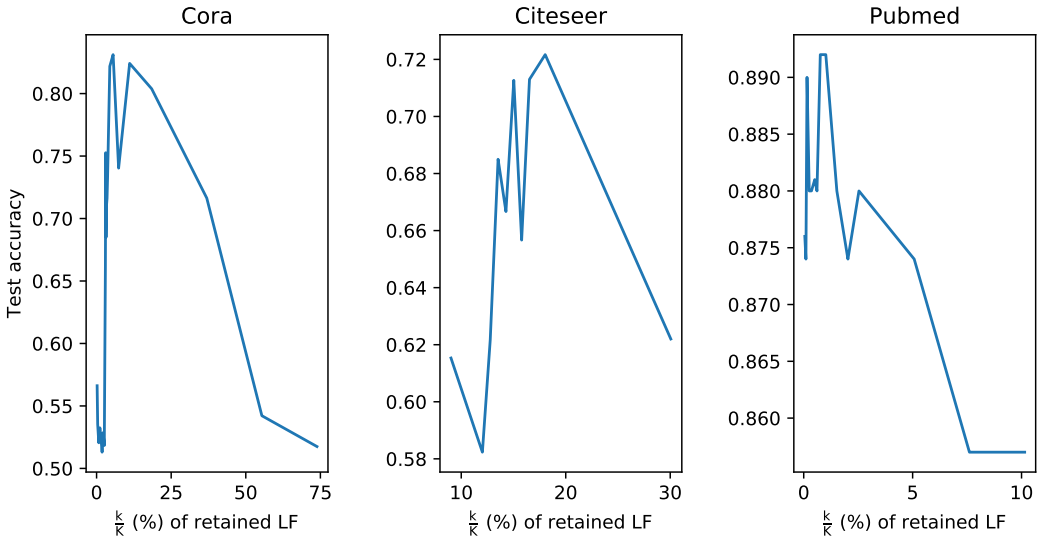

Figure 4: Accuracy of MLPs trained on $\tilde{x}_k$, according to the Low-Frequency band $[\lambda_k, \lambda_0]$ retained.

racy drops, which indicates that high-frequencies behave like a residual noise that is not well filtered by a MLP, and rather overfitted. This is particularly true for Pubmed, which has a fast spectral decay and for which the low frequencies seem to be the most informative.

Table 1: Comparison of various models on Cora, Citeseer and Pubmed.

Method	Data augmentation	Cora	Citeseer	Pubmed
GCN (Rong et al., 2019b)	No	**86.6**	**79.3**	90.2
(Chen et al., 2018)	No	86.5	-	88.8
MLP (ours) on x	No	74.0	73.3	89.1
MLP (ours) on $\tilde{x}_k$ (ours)	No	**86.6**	77.3	**91.4**
DropEdge (Rong et al., 2019b)	Yes	88.2	80.5	91.7
(Huang et al., 2018)	Yes	87.4	79.7	90.6

Boosting MLP performances We perform a hyper-parameter grid search on each dataset in order to investigate MLPs strength further, and report the case giving the best accuracy on each benchmark validation set. We summarize our findings in Tab. 1. In particular, one uses a fraction 5.9%, 15.0% and 0.7% of the spectrum respectively on Cora, Citeseer and Pubmed in order to obtain our best performances. We note that a MLP trained solely on x already outperforms the approach of Chen et al. (2018), without relying on a graph structure. More details on the methodology are provided in the supplementary material. This simple model is highly competitive with concurrent works like Rong et al. (2019b); Huang et al. (2018); Chen et al. (2018), and in particular with vanilla GCNs. Note also that our method does not incorporate any data augmentation procedure such as Rong et al. (2019a), and thus a performance gap still remains. Data augmentation methods could also be potentially incorporated, at the price of an extended computation time.

Computational complexity remark The MLPs introduced above are of interest if the corresponding graph topology is fixed, with a large graph, and high connectivity. Indeed, using a MLP allows to easily employ mini-batch strategies and the training data can be linearly reduced according to the fraction of low frequencies being kept. On the other hand, it requires computing a truncated SVD which can be computationally expansive despite existing efficient approximation algorithms (Coutino et al., 2020), depending on the number of eigenvalues needed. (however, computing the first hundred eigenvectors takes few second on datasets like Citeseer.

4.2 HIGH-FREQUENCY ABLATIONS

4.2.1 STABILITY TO HIGH FREQUENCIES

We now study the stability of a GCN w.r.t. spectral variations. In the case of image processing, it is standard that low-frequencies almost do not affect the classification performances and that perturbations of high-frequencies lead to instabilities. We would like to validate this principle, and to do so, we consider $\nabla_\Lambda \ell$, which is the gradient w.r.t. every singular value λ_k. Small amplitudes of $|\frac{\partial \ell}{\partial \lambda_k}|$ indicate more stable coefficients. Fig. 5 plots the amplitude of the gradient w.r.t. λ_k at the initialization of a GCN and after training. First, we note that a GCN is more sensitive to spectral perturbations after training, which is logical because the GCN adapts its weights to the specific structure of a given graph. However, after training, we remark that the high-frequency perturbations have a small impact compared to the low-frequency perturbations on the three datasets, except for Pubmed which has dominant low-frequencies. This is consistent with the findings of the previous section.

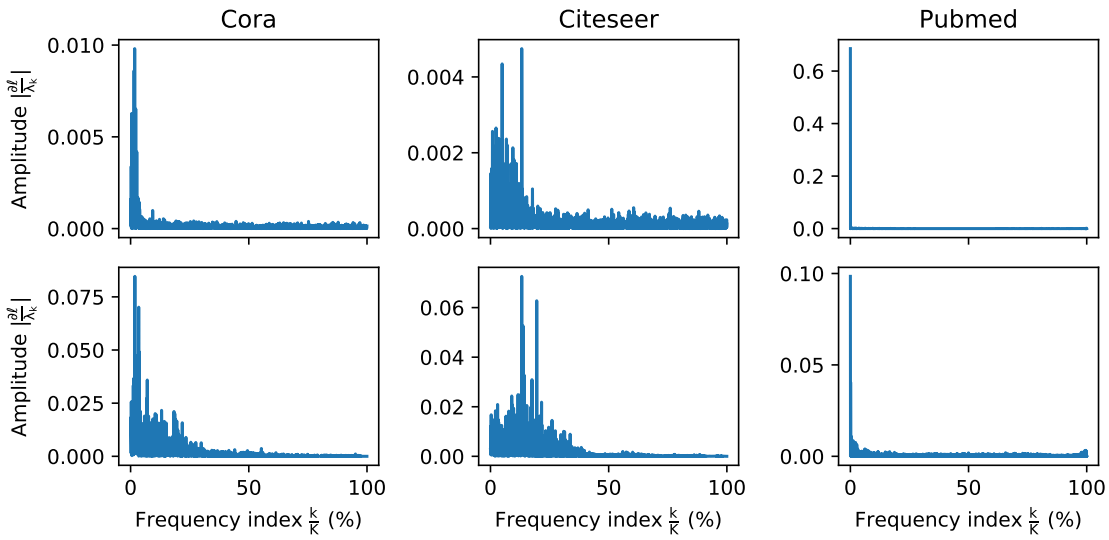

Figure 5: The (top) and (bottom) figures corresponds to $|\frac{\partial \ell}{\lambda_k}|$ of the same model taken respectively at the initialization and the end of training.

4.2.2 SMOOTHNESS AND SELF-LOOPS

As proposed by Kipf & Welling (2016), we now study how the self-loop of A affects our GCN training procedure and in particular, we study it through the lens of spectral analysis. To do so, we employ $\Pi_\eta(\lambda) = \frac{2\lambda+\eta-1}{1+\eta}$, which leads to:

$$A_\eta \triangleq \Pi_\eta(A) = \frac{1}{1+\eta}\left(\eta \boldsymbol{I} + \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\right), \tag{7}$$

where η can be understood as a smoothness parameter. Note that for $\eta < 1$, the spectrum of A_η is not necessarily positive and that $\|A_\eta\| = 1, \forall \eta$. Here, A_η enhances the high-frequencies of the averaging while allowing to balance a trade-off between smoothing and high-frequency selection, which can be here observed as,

$$A_\eta \underset{\eta\to\infty}{\sim} \boldsymbol{I} \qquad \text{and} \qquad A_0 = \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}. \tag{8}$$

As observed in Li et al. (2018), oversmoothing phenomena occur because one propagates an input through a cascade of n smoothing operators. Here, we note that a simple taylor expansion reveals

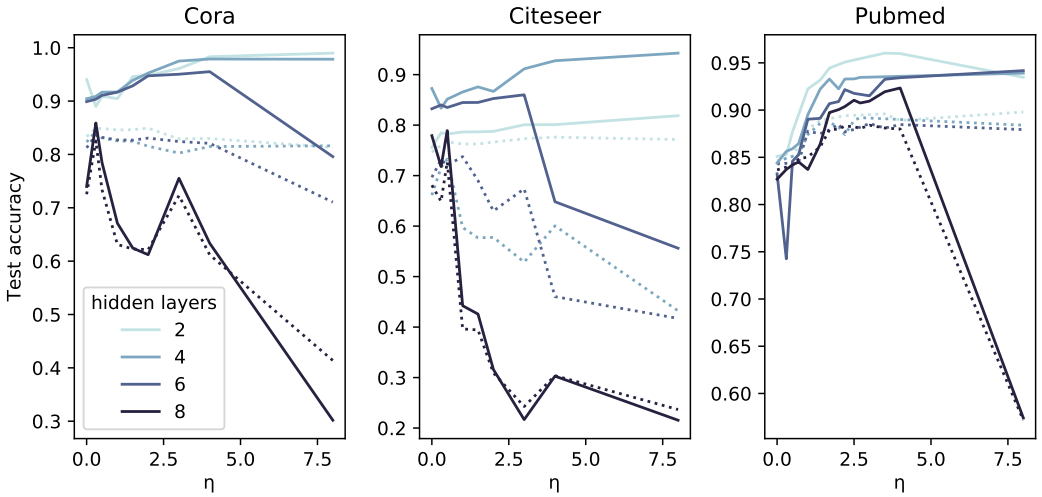

Figure 6: Test (dashed line) and train (plain line) accuracy according to η for various depth of a GCN.

asymptotically this trade-off:

$$A_\eta^n = \sum_{\lambda_k=1} e_k e_k^T + \sum_{\lambda_k<1} e_k e_k^T - \frac{2n(1-\lambda_k)}{\eta} e_k e_k^T + o(\frac{1}{\eta^2}) \,, \tag{9}$$

Thus depending on the nature of e_k, the degree of averaging can be adjusted.

Fig. 6 indicates that it is possible to adjust the degree of oversmoothing in order to train deeper networks ($\eta < 1$) and to slightly boost the accuracy of a GCN. We report both the training and testing accuracies in order to emphasize the impact of η on the training dynamics of a GCN. Citeseer exhibits a significant generalization gap for large value of η and a depth of 4: high-frequencies deteriorate the generalization properties. We note that varying η allows to improve the training accuracy of deeper GCNs, though larger η emphasizes higher-frequencies which leads to a significant drop in accuracy. We finally observe that selecting η via the best accuracy on a validation set allows minor boosts in performances: of the order of 1.4% ($\eta = 3$) and 1.5% ($\eta = 2$) for a GCN of depth 2 respectively on Citeseer and Pubmed.

5 Conclusion

In this work, we have studied the classification performance of a GCN if applying low-pass and high-pass filters. Our finding is that by design, a GCN mainly uses the low-frequencies, and that the role of the high-frequencies contrasts with their standard use in signal processing. Then, we are able to design MLPs that rely simply on a few eigenvectors of the graph Laplacian that are competitive with supervised graph approaches. We also study the stability of a GCN w.r.t. spectral perturbations, and show that they are more robust to high-frequency, contrary to CNNs.

Our work indicates that not only more difficult graph benchmarks are necessary, but also benchmarks that whose optimal model would rely on the high-frequencies of a graph Laplacian. It also shows that standard GSP tools such as graph wavelets might be simplified to low-pass filters, as we observed that the high-frequency does not bring a significant amount of information, and can even be interpreted as a residual noise.

REFERENCES

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pp. 1626–1633. IEEE, 2011.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. *CoRR*, abs/1801.10247, 2018. URL `http://arxiv.org/abs/1801.10247`.

Mario Coutino, Sundeep Prabhakar Chepuri, Takanori Maehara, and Geert Leus. Fast spectral approximation of structured graphs with applications to graph filtering. *Algorithms*, 13(9):214, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pp. 3844–3852, 2016.

Shay Deutsch and Stefano Soatto. Spectral embedding of graph networks, 2020.

Vishwaraj Doshi and Do Young Eun. Fiedler vector approximation via interacting random walks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–28, 2020.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Diffusion scattering transforms on graphs. *arXiv preprint arXiv:1806.08829*, 2018.

Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. *arXiv preprint arXiv:1905.04497*, 2019.

Feng Gao, Guy Wolf, and Matthew Hirn. Geometric scattering for graph data analysis. In *International Conference on Machine Learning*, pp. 2122–2131, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL `http://arxiv.org/abs/1608.06993`.

Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in neural information processing systems*, pp. 4558–4567, 2018.

Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Tackling over-smoothing for general graph convolutional networks, 2020.

Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. Convergence and stability of graph convolutional networks on large random graphs, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, pp. 13354–13366, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv preprint arXiv:1801.07606*, 2018.

Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. *arXiv preprint arXiv:1802.07510*, 2018.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Nikhil Mehta, Lawrence Carin, and Piyush Rai. Stochastic blockmodels meet graph neural networks. *arXiv preprint arXiv:1905.05738*, 2019.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Edouard Oyallon. Interferometric graph transform: a deep unsupervised graph representation. In *Proceedings of Machine Learning and Systems 2020*, pp. 4124–4134. 2020.

Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. The truly deep graph convolutional networks for node classification. *CoRR*, abs/1907.10903, 2019a. URL `http://arxiv.org/abs/1907.10903`.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2019b.

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

Ke Sun, Piotr Koniusz, and Zhen Wang. Fisher-bures adversary graph convolutional networks. In *Uncertainty in Artificial Intelligence*, pp. 465–475. PMLR, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Saurabh Verma and Zhi-Li Zhang. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1539–1548, 2019.

Felix Wu, Amauri H Souza Jr, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.

Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting" over-smoothing" in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1399–1407, 2019.

A APPENDIX

A.1 DATASET STATISTICS

Table 2: Dataset Statistics

Datasets	Nodes	Edges	Classes	Features	Traing/Validation/Testing split
Cora	2,708	5,429	7	1,433	1,208/500/1,000
Citeseer	3,327	4,732	6	3,703	1,812/500/1,000
Pubmed	19,717	44,338	3	500	18,217/500/1,000

A.2 HYPER-PARAMETER DESCRIPTION

Table 3: Hyper-parameter Description

Hyper-parameter	Description
lr	learning rate
hidden layers	the number of hidden layers
weight-decay	L2 regulation weight
dropout	dropout rate
frequencies	the number of low frequencies to add
eigenvector features normalization	whether to normalize the new MLP features in line (per nodes) or column (per features)

A.3 ON THE CHOICE OF THE NORMALIZATION

GCNs defined in Kipf & Welling (2016) don't exactly used the first order Laplacian approximation, but introduce a normalization trick : $A = (D+\boldsymbol{I})^{-\frac{1}{2}}(\mathcal{A}+\boldsymbol{I})(D+\boldsymbol{I})^{-\frac{1}{2}}$. Using the standard hyper-parameters defined in Section 4, we averaged our results on 3 seeds and observed no significant difference across datasets, as shown in Fig. 7.

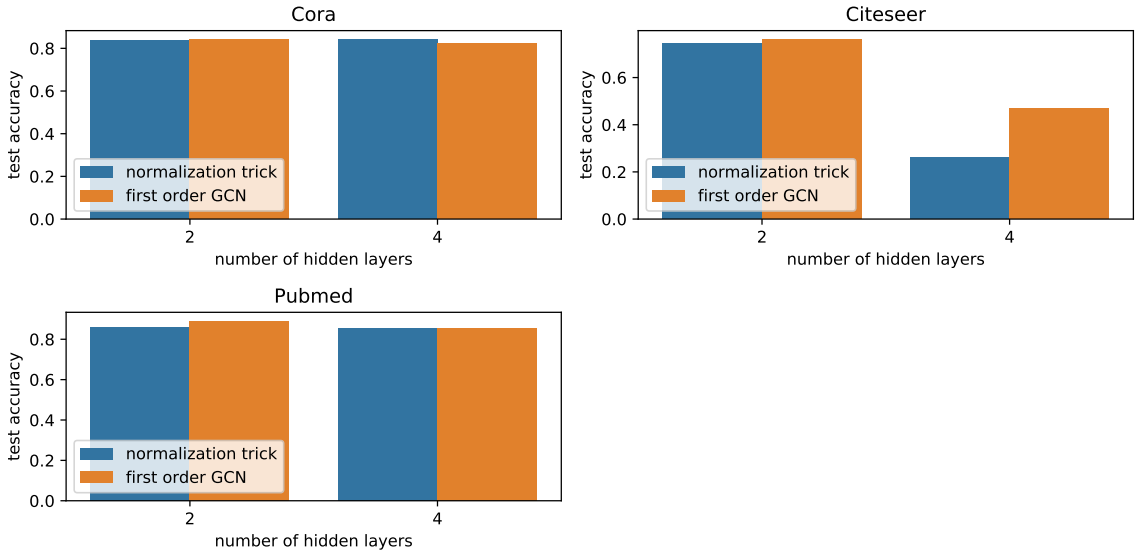

Figure 7: Comparison between the standard Laplacian first order normalization and the GCN normalization trick