universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

On transferability of graph neural networks

verfasst von | submitted by

David Crespo Cuaresma

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 821
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Mathematik
Betreut von | Supervisor:	Assoz. Prof. Dr. Philipp Christian Petersen M.Sc.

Abstract

Diese Arbeit zielt darauf ab, neueste Ergebnisse zur Übertragbarkeit von Graph-Neuronalen Netzwerken zusammenzustellen. Zu Beginn werden klassische Literatur und Methoden überprüft, um ein Verständnis der zugrunde liegenden Ansätze zu erlangen. Anschließend werden spektrale Graph-Neuronale Netzwerke und Werkzeuge der Funktionalanalysis verwendet, um zu beweisen, dass Filter, die aus Funktionen im Cayley-Glattheitsraum gebildet werden, tatsächlich übertragbar sind. Im weiteren Verlauf untersuchen wir einen neueren Ansatz unter Verwendung von Graphons. Durch die Definition von Grenzobjekten von Graphen stellen wir fest, dass Filter, die unter Verwendung von Graphons definiert werden robust sind, wenn Glattheitsbeschränkungen der Aktivierungsfunktionen des jeweiligen neuronalen Netzwerks vorliegen. Diese Arbeit schließt mit möglichen Ausgangspunkten für neue Forschungsansätze im aktuellen maschinellen Lernen.

Abstract

This thesis aims to compile recent results in the transferability of graph neural networks. At the start we review classical literature and methods in order to gain an understanding of the underlying methods. Then, by using spectral graph neural networks and tools from functional analysis, we prove that using filters, made from functions inside the Cayley smoothness space, are in fact transferable. Subsequently, we investigate a newer approach using graphons. By defining limit objects of graphs, we find that filters defined using graphons are especially robust, given smoothness restrictions of the activation functions of the given neural network. This thesis concludes by giving possible starting points for new research in current machine learning.

Contents

1 Introduction

In the past years, interest in machine learning has steadily increased. As machine learning becomes more accessible, it is important that the algorithms used react similarly to similar situations. For example, in personalized algorithms two people with similar interests should be recommended similar new hobbies to try or perhaps an algorithm that recommends products should be able to give reasonable advice for similar products. We call such algorithms transferable and it is the property we investigate in this thesis. Graph neural networks are particularly interesting in this regard as a lot of data can be encoded into graphs, such as street networks or online article affiliations. This thesis aims to investigate the current literature on transferability of graph neural networks.

To start the thesis we set the foundations for the topics discussed, which then leads nicely into the topics presented in [Levie et al., 2019]. In this section about spectral methods, we investigate graph filters that primarily use the properties of Fourier analysis. We introduce the so called Cayley-smoothness space in order to prove that for a certain class of functions graph neural networks are transferable.

In the following sections, this thesis presents another approach used in graph neural networks. By introducing a type of "continuous graph" which is called graphon, one can use many of the tools already known from functional analysis in order to analyse graph neural networks. This is a complex and young field of research. This part of the thesis mainly focuses on the results presented in [Maskey et al., 2023]. To start we lay down the definitions for graphons and how we generalise graph filters to graphon filters. [Maskey et al., 2023] also admits that not all graph shift operators can be represented by graphon shift operators, so a new class of shift operators, so called unbounded shift operators are introduced here to overcome this gap. The paper closes with the transferability results about graphons given in [Maskey et al., 2023].

2 Historical background

Investigating the properties of graphs is a well established field. Mathematicians have been trying to understand graphs since 1736, when Leonard Euler first published a paper on the now famous problem of the seven bridges in Königsberg. However, the word Graph was only used much later, in a 1878 paper by Sylvester. Since then, the field has evolved, spanning from using graphs in order to calculate maximal flows through systems of pipes to using so called random graphs to describe random processes.

In the 19th century there was no access to methods such as machine learning. This means that often one had to rely on human intuition in order to find clusters of points. Generally, this is a good approach since humans have a good intuition when it comes to these problems, however intuition might fail when visualising the data set is troublesome. An example of this are data sets where, if there are more than three labels per data point visualising becomes difficult, giving rise to the methods discussed below.

2.1 Classical methods

This section, will briefly discuss methods of clustering, which are unsupervised machine learning algorithms in order to find clusters in a mass of data points.

Linkage based clustering as in [Shalev-Shwartz and Ben-David, 2014] Section 22.1 is an intuitive method that quickly produces clusters (i.e. collections of points that are close in a given norm). We shall briefly explain this process informally, as the concept is very intuitive. The first step is to define the initial clusters, which are very simply put clusters only containing one data point. Afterwards we merge similar clusters, but we need to define what similar means. Often our clusters are in some metric space, so we can use the distance between the initial clusters to form the new ones. After merging the first clusters one can repeat this process until one reaches the desired number of clusters or until only one cluster containing remains.

Another simple way to find clusters is via so called k-means clustering (found in [Shalev-Shwartz and Ben-David, 2014] Section 22.2). The gist of this method is that one wants to find k clusters, such that the data points of a given cluster are as close to the center of said cluster as possible. To achieve this there exists an algorithm called Lloyd's algorithm, which iteratively finds the centres of the clusters. However, as useful as this algorithm is, it does not always converge or sometimes converges to a non-optimal clustering.

Lastly, we can mention spectral clustering ([Shalev-Shwartz and Ben-David, 2014] Section 22.3). We shall not discuss this method in detail since it is almost a precursor to some of the methods presented further below. In spectral clustering one divides a graph into subgraphs by minimizing the so called cut of the graph. The convenient thing about spectral clustering is that it also works for non-convex data sets, for which the previous two methods have average or bad results.

3 Preliminaries

This section will lay down basic definitions that we will use throughout the thesis and explain some of the basic notation.

3.1 Graphs and graph Laplacians

Definition. ([Shalev-Shwartz and Ben-David, 2014] section 22.3) An undirected graph G is a tuple consisting of two sets: V the set of vertices which are usually referred to as $v_1, v_2, \dots v_n$ for $n \in \mathbb{N}$ and a set of tuples which signify the edges $E \subset V \times V$. We say that two vertices v_i and v_j are connected if $(v_i, v_j) \in E$. Furthermore, we define the adjacency matrix of a graph G as

$$(A)_{i,j} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Obviously, the adjacency matrix of an undirected graph is symmetric, since we assume an undirected graph therefore $((v_i, v_j) \in E) \leftrightarrow ((v_j, v_i) \in E)$.

In real life applications, using undirected graphs to represent real life circumstances is often intuitive, like mapping out a city with bus lines travelling through. However, at given times some streets might be busier than others. In order to represent that phenomenon, we introduce weights on a given edge that represents such a concept.

Definition. ([Shalev-Shwartz and Ben-David, 2014] Section 22.3) A weighted graph is a triple $G = (V, E, W)$ where (V, E) defines an undirected graph and $W \in \mathbb{R}^{n \times n}$ is a symmetric matrix with positive entries and $n = |V|$.

We can now use this new notion of graph to introduce the graph Laplacian. Oftentimes this special matrix is used for spectral clustering.

Definition. ([Shalev-Shwartz and Ben-David, 2014] Section 22.3) We define the degree matrix D of an undirected weighted graph $G = (V, E, W)$ by the sum

$$D_{i,i} = \sum_{(i,j) \in E} w_{(i,j)}. \tag{3.2}$$

Here $w_{(i,j)}$ denotes the entries of the weight matrix W.

Definition. ([Shalev-Shwartz and Ben-David, 2014] Section 22.3) Let $G = (V, E, W)$ be a graph with weight matrix W. The graph Laplacian of an undirected weighted graph is defined as $L = D - W$, where D is the degree matrix of G.

The graph Laplacian is also our first example of a so called shift operator, which we will introduce later when we cover spectral graph filters. Furthermore, in the following we will call a function f from the set of vertices of a graph into $\mathbb{C}$ a graph signal.

3.2 Neural networks

For convenience we introduce the basic structure of shallow neural networks, since they are the basis of all other neural networks.

Definition. [Hastie et al., 2009] We call a function $f : \mathbb{R}^d \to \mathbb{R}$ of the following form neural network.

$$f(x) = \sum_{i=1}^{N} c_i \rho(\langle a_i, x \rangle + b_i) + r. \tag{3.3}$$

We say that f has input dimension $d \in \mathbb{N}$, $N \in \mathbb{N}$ neurons, the activation function $\rho : \mathbb{R} \to \mathbb{R}$, the weights $a_i \in \mathbb{R}^d, c_i \in \mathbb{R}$ and biases $b_i \in \mathbb{R}$ and $r \in \mathbb{R}$ for each $i \in [N]$.

In practice, one would usually use deep neural networks for one's machine learning task, which introduce so called hidden layers, in which each neuron applies a linear transformation to the given input vector before applying the scalar product and activation function. Neural networks are a powerful class of approximating functions since they can approximate every continuous function on a compact set as long as the activation function is sigmoidal (this means the activation function is continuous, has $lim_{x\to-\infty}\sigma(x) = 0$ and $lim_{x\to\infty}\sigma(x) = 1$)[Cybenko, 1989]. As we can see from the definition, neural networks are usually defined on the real numbers, so we need to do some work to define them on graphs. This thesis will confine itself to graph convolutional networks (GCNNs). To understand these, we must first define what a convolutional neural network is.

Definition. (informal) A convolutional neural network is a neural network comprised of 3 special kinds of layers. Convolutional layers which operate just as discrete convolution, pooling layers, where the information is compressed, and fully connected layers, which operate like the ones we are already familiar with, from deep neural networks.

Graph convolutional neural networks (GCNNs) are a special kind of neural network that use special kinds of convolutional layers to translate graphs into a structure that neural networks can understand.

Definition. (informal) A graph convolutional neural network is a CNN that instead of being defined on $\mathbb{R}$ is defined on graphs. It uses the same layers

as regular CNNs with the main change that in the convolutional layer we can not use the usual convolution. Instead graph convolutional networks perform convolution on the spectrum on the graph Laplacian, which is given by the eigenvectors of the graph Laplacian.

Now that we understand the object that we are working with, we give a notion of what the aim of this thesis is.

3.3 Definition of transferability

The goal of this thesis is to show that graph filters that are commonly used for GCNNs are transferable, which in a very informal sense means that a GCNN with a transferable filter that is trained on similar training sets will have similar results.

Definition. We call a function $f : \mathbb{R}^n \to \mathbb{R}^m$ equivariant if for all permutations of n elements σ we have that $f(x_{\sigma(1)}, \dots, x_{\sigma(n)}) = f(x_1, \dots, x_n)$.
Let $f : \mathbb{R}^n \to \mathbb{R}^m$. We call f stable if for all $\varepsilon > 0$ there is $\delta > 0$ such that for $x, E \in \mathbb{R}^n$ and $y = x + E$, where $||E|| < \delta$, there is $\varepsilon > 0$ such that $||f(x) - f(y)|| < \varepsilon$.
A function that is both stable and equivariant is called transferable.

We also investigate asymptotic transferability, which means that even if we can not find bounds for stability, we can still guarantee convergence of the difference between the approximation and the correct function.

4 Graph filters

Originally, filters are used in functional calculus, so we must first define what it means to use filters on graphs. For that, we introduce graphs with graph shift operators (GSO) and then define the notion of a graph filter.

4.1 General graph filters

Definition. [Levie et al., 2019] A graph shift operator is a self adjoint operator. A graph with GSO is a graph together with some GSO.

Usually one uses a combinatorial object as the GSO of a graph. Popular and intuitive choices are the Laplacian and the normalized Laplacian $L_N = I - D^{1/2}WD^{1/2}$, where D is the degree matrix of G and W is the weight matrix of G. However, one can even construct a GSO from a given data set. As we will see, choosing a reasonable GSO is imperative.

Definition. [Maskey et al., 2022] Let G be a graph with a GSO Δ, and let $\{\lambda_i, \phi_i\}_{i=1}^n$ be the eigenvalues and eigenvectors of Δ. We call a continuous function $h : \mathbb{R} \to \mathbb{R}$ a filter and define the operator $h(\Delta)$ by $h(\Delta)x = \sum_{i=1}^n h(\lambda_i) \langle x, \phi_i \rangle \phi_i$. Here $\langle ., . \rangle$ denotes the standard scalar product. We call $h(\Delta)$ the realization of the filter but for brevity we will often use it interchangeably with the word filter.

To illustrate the concept, we give an example for a very simple filter.

Example. [Maskey et al., 2022] Let h be a rational funtion, i.e. h is of the form

$$h(\lambda) = \frac{\sum_{n=0}^N h_n \lambda^n}{\sum_{m=0}^M h_m \lambda^m}. \tag{4.1}$$

A simple calculation shows that the realization of this filter is given by

$$h(\Delta) = \left(\sum_{n=0}^N h_n \Delta^n \right) \left(\sum_{m=0}^M h_m \Delta^m \right)^{-1}. \tag{4.2}$$

This means that we can use rational functions and polynomial functions to build filters. [Defferrard et al., 2016]

4.2 Spectral graph filters

In order to introduce spectral graph filters we need to introduce what it means to apply the Fourier transformation on a graph signal, since usually graph signals are functions on a discrete space instead of continuous functions.

Definition. [Levie et al., 2019] Let $G = (V, E, \Delta)$ be a graph with GSO and $f : V \to \mathbb{C}$ be a graph signal. Then we define the Fourier transform of a graph signal by $\mathcal{F}f = (\langle f, \phi_n \rangle)_{n=1}^N$ where ϕ_n are the eigenvectors of Δ.

From this we then construct our first spectral graph filter.

Definition. [Levie et al., 2019] Let $G = (V, E, \Delta)$ be a graph with GSO and $f : V \to \mathbb{C}$ be a graph signal. We define a spectral graph filter $\mathcal{G}$ with coefficients $(g_n)_{n=1}^N$ by $\mathcal{G}f = \left(\sum_{n=1}^N g_n \langle f, \phi_n \rangle \phi_n \right)$.

As one can clearly see, this filter is equivariant, since it only depends on the scalar product of the eigenvectors, which is equivariant, because the scalar product does not care about the order of components of the eigenvectors. This is already half of what we need for transferability. However these filters have two glaring issues. First, since they require the eigenvalue decomposition of the GSO, they are computationally costly. However, the fatal flaw is that spectral filters defined in this sense are not transferable.

Example. [Levie et al., 2019] Let $G = (V, E, \Delta)$ be a graph with GSO. Here Δ is the Laplacian. A small perturbation in Δ can cause a large perturbation in the eigenvalue decomposition of Δ, meaning using this method leads to non-transferability.

However, one can prevent those shortcomings by taking an approach that is based in functional analysis:

Definition. [Levie et al., 2019] Let $g : \mathbb{C} \to \mathbb{C}$ be a function and T a self-adjoint unitary operator with finitely many eigenvalues λ and eigenvectors ϕ_i on the Hilbert space H. Let $f \in H$, then we define the operator $g(T)$ as

$$g(T)f = \sum_{i=1}^{n} g(\lambda_i) \langle f, \phi_i \rangle \phi_i. \tag{4.3}$$

By using this decomposition, it is no longer necessary to compute the eigenvalues of the GSO or any Fourier coefficients. Even inversion is simple, since it just comes down to solving a system of linear equations. Many computationally efficient methods are known for solving those already.

5 Graph convolutional neural networks under spectral filters

In this section, we apply spectral filters to GCNNs and investigate their transferability. It has been shown that filters are transferable if and only if they are equivariant (i.e. renaming the indices of the data does not change the output) and stable (i.e. using "similar" graphs as inputs produced "similar" outputs).

5.1 Transferability result for spectral filters

It is easy to see why spectral filters in the above sense are equivariant under renaming of the vertices, as they only depend on the scalar product of the eigenvectors, which is equivariant. Therefore, we already proved the first of two properties of transferability. We prove stability on the so called Cayley smoothness space.

Definition. [Levie et al., 2019] Let $e^{i\mathbb{R}}$ define the unit circle and $\mathcal{C} : \mathbb{R} \to e^{i\mathbb{R}}$ be the Cayley transform $\mathcal{C}(x) = \frac{x-i}{x+i}$. The Cayley smoothness space $Cay^1(\mathbb{R})$ is the subspace of functions $g \in L^2(\mathbb{R})$ of the form $g(\lambda) = q(\mathcal{C}(\lambda))$ where $q : e^{i\mathbb{R}} \to \mathbb{C}$ is in $L^2(e^{i\mathbb{R}})$, and has classical Fourier coefficients $(c_l)_{l=1}^{\infty}$ satisfying $||g||_{\mathcal{C}} := \sum_{l=1}^{\infty} l|c_l| < \infty$.

Now we prove the stability result for filters based on functions from this space.

5.1.1 Stability result for spectral filters

In order to prove stability, let us first recall two useful lemma from spectral theory about self adjoint matrices and bounded normal operators.

Lemma 5.1. *[Levie et al., 2019] Suppose $B, D, E \in \mathbb{C}^{N\times N}$ are self adjoint matrices satisfying the relation $B = D + E$ and $||B||, ||D|| < C$ for $C > 0$ then for every $l \geq 0$ we have*

$$||B^l - D^l|| \leq lC^{l-1}||E||. \tag{5.1}$$

Lemma 5.2. *[Levie et al., 2019] Let T be a bounded normal operator in a Hilbert space. Let σ be the spectrum of T. Define the infinity norm on the space of bounded continuous functions $f : \sigma \to \mathbb{C}$ by*

$$||f - g||_\infty^\sigma = \sup_{x\in\sigma} |f(x) - g(x)|. \tag{5.2}$$

Then

$$||f(T) - g(T)|| = ||f - g||_\infty^\sigma. \tag{5.3}$$

Here $||.||$ denotes the operator norm.

For stability [Levie et al., 2019] proves the following bound:

Theorem 5.3. *[Levie et al., 2019] Let $\Delta \in \mathbb{C}^{N\times N}$ be a self-adjoint matrix. Let $\Delta' = \Delta + E$ be self-adjoint, such that $||E|| < 1$. Let $g \in Cay^1(\mathbb{R})$. Then*

$$||g(\Delta) - g(\Delta')|| \leq ||g||_c \left((||\Delta|| + 1)\frac{||E||}{1 - ||E||} + ||E|| \right). \tag{5.4}$$

Proof. The proof presented here follows the steps from [Levie et al., 2019]. First, we restrict the statement onto a dense subset. Second we extend the estimation to the full space. The dense subspace of $Cay^1(\mathbb{R})$ we choose is $g = q \circ \mathcal{C} \in Cay^1(\mathbb{R})$ where q has a finite expansion with coefficients $(c_l)_{l=1}^L$.
First, let us note, that by definition of $\mathcal{C}$, we have

$$\mathcal{C}(\Delta) - \mathcal{C}(\Delta') = (\Delta - i)(\Delta + i)^{-1} - (\Delta - i)(\Delta' + i) + (\Delta - i)(\Delta' + i) - (\Delta' - i)(\Delta' + i), \tag{5.5}$$

therefore by taking the norm we get

$$||\mathcal{C}(\Delta) - \mathcal{C}(\Delta')|| \leq ||(\Delta - i)|| \left|\left|(\Delta + i)^{-1} - (\Delta - i)\right|\right| + \left|\left|(\Delta' + i)^{-1}\right|\right| ||E||. \tag{5.6}$$

Since Δ' has a real spectrum, which follows from the spectral theorem and being

self-adjoint, it follows that

$$||\Delta' + i||^{-1} \leq (||\Delta + E|| + 1)^{-1} \leq (||\Delta|| + ||E|| + 1)^{-1} \leq \tag{5.7}$$

$$\left(\sup_{\lambda \in \sigma(\Delta)} (||\lambda||) + 2 \right)^{-1} \leq 1. \tag{5.8}$$

Here $\sigma(\Delta)$ denote the spectrum of Δ, which is bounded since Δ has at most n eigenvalues. We have

$$||\mathcal{C}(\Delta) - \mathcal{C}(\Delta')|| \leq (||\Delta|| + 1)||(\Delta + i)^{-1} - (\Delta' + i)^{-1}|| + ||E||. \tag{5.9}$$

Now we bound $||(\Delta + i)^{-1} - (\Delta' + i)^{-1}||$. Here we use that $||E|| < 1$ and the Neumann series expansion in order to achieve:

$$(\Delta + i + E)^{-1} = (\Delta + i)^{-1}(I + E(\Delta + i)^{-1})^{-1} \tag{5.10}$$

$$= (\Delta + i)^{-1} \left(\sum_{k=0}^{\infty} (-1)^k \left(E \left(\Delta + i \right)^{-1} \right)^k \right). \tag{5.11}$$

Since $||(\Delta + i)^{-1}|| < 1$ we get

$$||(\Delta + i)^{-1} - (\Delta' + i)^{-1}|| \leq \frac{||E||}{1 - ||E||} \tag{5.12}$$

Combining (5.12) and (5.9) yields

$$||\mathcal{C}(\Delta) - \mathcal{C}(\Delta')|| \leq (||\Delta|| + 1)\frac{||E||}{1 - ||E||} + ||E||. \tag{5.13}$$

Notice that $\mathcal{C}(\Delta)$ and $\mathcal{C}(\Delta')$ are unitary, meaning their spectra are bounded by 1. Using Lemma 5.1, the triangle inequality, and polynomial expansion of $p(\mathcal{C}(\Delta)) - p(\mathcal{C}(\Delta'))$ we obtain the result for the truncated functions,

$$||g(\Delta) - g(\Delta')|| = ||p(\mathcal{C}(\Delta)) - p(\mathcal{C}(\Delta'))|| \leq \sum_{l=1}^{L} l c_l ||\mathcal{C}(\Delta) - \mathcal{C}(\Delta')||. \tag{5.14}$$

We use a density argument to show the full theorem. For a given $g = q \circ \mathcal{C}$, we define the truncation $g_L = q_L \circ \mathcal{C}$ by restricting the Fourier series of q to the coefficients $(c_l)_{l=1}^{L}$. Now we use the triangle inequality. Given $L \in \mathbb{N}$,

$$||g(\Delta) - g(\Delta')|| \leq ||g(\Delta) - g_L(\Delta)|| + ||g_L(\Delta) - g_L(\Delta')|| + ||g_L(\Delta') - g(\Delta')||. \tag{5.15}$$

The first and third term converge to 0 for large L, meaning we only need to account for the second term. For every $L \in \mathbb{N}$ we have

$$||g_L||_{\mathcal{C}} = \sum_{k=0}^{L} l|c_l| \leq \sum_{k=0}^{\infty} l|c_l| = ||g||_{\mathcal{C}}. \tag{5.16}$$

Therefore, for any $\varepsilon > 0$ we get

$$||g(\Delta) - g(\Delta')|| \leq ||g||_{\mathcal{C}} \left((||\Delta|| + 1) \frac{||E||}{1 - ||E||} + ||E|| \right) + \varepsilon. \tag{5.17}$$

Since ε is arbitrary, we get the result (5.4).

□

We would now like to give examples for functions in the Cayley smoothness space.

Example. Let $f : \mathbb{R} \to \mathbb{R}$ with the following estimates for its Fourier coefficients $c_n \leq n^{-1-\delta}$ for all $n \in \mathbb{N}$ with $\delta > 1$. First we show that f is in $L^2((-\pi, \pi))$ by using Parseval's identity ([Suetin, 1979]).

$$||f||_{L^2((-\pi,\pi))} = 2\pi \sum_{n=0}^{\infty} |c_n|^2 \leq 2\pi \sum_{n=0}^{\infty} (n^{-1-\delta})^2 \leq \infty. \tag{5.18}$$

Therefore, we have that $f \in L^2(-\pi, \pi)$. Now we prove that f is indeed in $Cay^1((-\pi, \pi))$.

$$||f||_{\mathcal{C}} = \sum_{n=0}^{\infty} n|c_n| \leq \sum_{n=0}^{\infty} n n^{-1-\delta} = \sum_{n=0}^{\infty} n^{-\delta} \leq \infty. \tag{5.19}$$

The sum remains bounded since $\delta > 1$. We conclude that $f \in Cay^1(-\pi, \pi)$.

6 Graphons

As mentioned above, we want to say define convergence for a sequence of graphs, we call the limit objects of these sequences graphons. We will make this notion precise below.

6.1 Definition

In order to define what a graphon is, we first establish some definitions for graphs and graph homomorphisms, since intuitively it is not entirely clear what it means for a graph to converge. In order to do that, let us define the homomorphism numbers, which count how many homomorphisms exist from one graph to another.

Definition. Let (G, E) and (H, F) be undirected graphs, we call $f : (G, E) \to (H, F)$ graph homomorphism if it preserves edges (i.e. $(v_i, v_j) \in E$ then $(f(v_i), f(v_j)) \in F$).

Example. One can think of graph homomorphisms as colouring each vertex that has the same value in one colour and then moving the vertices in 2d-space. At most one can add edges, since if one were to cut an edge it would no longer be a homomorphism. A popular example is the homomorphism from the flower snark graph J_5 into the cycle graph C_5. (see figure 1)

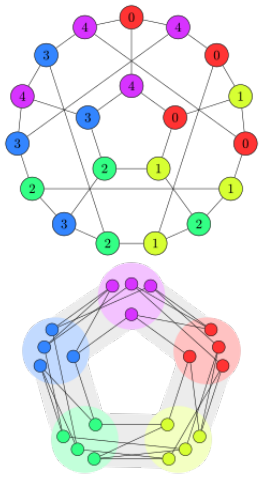

Figure 1: A visual representation of the homomorphism from the flower snark to the circle graph (By Tokenzero - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=57186561)

Definition. [Maskey et al., 2023] Let G a graph with GSO Δ. We define the GWM of G as $A = n\Delta$. Where $n \in \mathbb{N}$ is the number of nodes of G.

Definition. [Maskey et al., 2023] Let $F = (V(F), E(F))$ be a graph and G a graph with GSO Δ, GWM A and node set $V(G)$.
Given a map $\psi : V(F) \to V(G)$, we define the homomorphism number to be

$$hom_\psi(F, G) = \Pi_{(u,v)\in E(F)} A_{(\psi(u),\psi(v))}. \tag{6.1}$$

Here $A_{(\psi(u),\psi(v))}$ is the entry of A at $(\psi(u), \psi(v))$. We define the homomorphism number of two graphs as

$$hom(F, G) = \sum_{\psi: V(F)\to V(G)} hom_\psi(F, G), \tag{6.2}$$

the sum adds all the homomorphism numbers of each map ψ from F to G. Lastly the homomorphism density with respect to graph G is a function $t(\textbf{.}, G)$ that assigns to each graph F with weights 0 or 1 the following value

$$t(F, G) := \frac{|hom(F, G)|}{|V(G)|^{|V(F)|}}. \tag{6.3}$$

To give a brief explanation for homomorphism density, it describes the ratio of how many of the functions from graph F to graph G are homomorphisms. Naively, it is the probability of randomly drawing a homomorphism from the basket of all functions between the two graphs. For our purposes, we often have $F \subset G$ and then the homomorphism numbers give a quantity as to how often F appears as a structure in G.

Before we introduce what it means for a sequence of graphs to converge, let us introduce the formal definition of a graphon, which at first might seem almost arbitrary but is shown to be the natural limit of graphs.

Definition. [Maskey et al., 2023] A graphon is a bounded, symmetric and measurable function $W : [0, 1] \times [0, 1] \to \mathbb{R}$. We denote the space of all graphons by $\mathcal{W}$. Let $\Gamma > 0$. The subspace $\mathcal{W}_\Gamma \subset \mathcal{W}$ is defined as the set of all graphons $W \in \mathcal{W}$ such that $|W(u, v)| \leq \Gamma$ for every $(u, v) \in [0, 1] \times [0, 1]$. Additionally we define the graphon shift operator (GRSO) T_W as $T_W f(u) := \int_0^1 W(u, v) f(u) dv$.

It could be confusing that the limit of something discrete like a graph can be a function defined on continuous space, however the way that one should imagine graphons is as a space from which discrete graphs are sampled, such, one calls such a graphon latent space of graphs that correspond with it. To further generalise the notions above, let us next define, homomorphism density of a graphon.

Definition. [Maskey et al., 2023] For a graphon W and a graph $F = (V, E)$ with weights either 0 or 1, the homomorphism density of F in W is defined as

$$t(F, W) := \int_{[0,1]^n} \Pi_{(i,j)\in E} W(u_i, u_j) \Pi_{i\in V} du_i, \tag{6.4}$$

where n is the number of vertices in F.

When we talk about convergence of graphs, we actually usually talk about convergence in the sense of the above mentioned homomorphism density, so let us formalize this.

Definition. [Maskey et al., 2023] Let $(G_n)_n$ be a sequence of graphs with GSOs and GWMs $(A_n)_n$.

We say that $(G_n)_n$ is a bounded sequence if there exists a constant $C > 0$ such that $||A_n|| < C$ for all $n \in \mathbb{N}$.
We say that $(G_n)_n$ converges in homomorphism density, if for all graphs F the sequence $(t(F, G_n))_n$ converges.

Let us now summarize the above claims in the following theorem.

Theorem 6.1. *[Borgs et al., 2008] For every bounded sequence of graphs $(G_n)_n$ with GSO that converges in homomorphism density, there exists a graphon $W \in \mathcal{W}$ such that for every simple graphs F*

$$t(F, G_n) \to t(F, W) \tag{6.5}$$

for $n \to \infty$.

Now that we know that we can pass from sequences to graphs to graphons, perhaps it would be nice to also be able to pass from a graph to a so called induced graphon.

Definition. [Maskey et al., 2023] Let G be a graph with GSO Δ and GWM A and n nodes and $\{P_1, \dots, P_n\}$ a partition of $[0, 1]$. The graphon W_A induced by G is defined as

$$W_A(u, v) := \sum_{i,j \leq n} A_{(i,j)} \chi_{P_i}(u) \chi_{P_j}(v). \tag{6.6}$$

We also get that $t(F, G) = t(F, W_A)$ for graphs G with GWM A and every graph F with weights either 0 or 1.

Furthermore, we define convergence of graphs.

Definition. [Maskey et al., 2023] Let $(G_n)_n$ be a sequence of graphs with GSOs and corresponding GWMs $(A_n)_n$. Let $W \in \mathcal{W}$ satisfy $G_n \to W$. Let $(Perm_G)_n$ the set of permutations of vertices of G. The set of *admissible permutations* is defined as

$$\mathbf{P} = \left\{ (\pi_n)_n \in (Perm_{G_n})_n \mid \|T_{W_{\pi_n(A_n)}} - T_W\|_{L^2([0,1]) \to L^2([0,1])} \to 0 \text{ for } n \to \infty \right\}.$$

Let $(G_n)_n$ be a sequence of graphs with GSOs and $(\mathbf{x}_n)_n$ a sequence of graph signals defined on these graphs. We say that $(\mathbf{x}_n)_n$ converges to a graphon signal $\psi \in L^2([0, 1])$ if there exists a graphon W such that $G_n \xrightarrow[n\to\infty]{H} W$ and for some admissible permutation $(\pi_n)_n \in \mathbf{P}$

$$\|\psi_{\pi_n(\mathbf{x}_n)} - \psi\|_{L^2([0,1])} \xrightarrow{n\to\infty} 0. \tag{6.7}$$

We write $\mathbf{x}_n \xrightarrow[n\to\infty]{I} \psi$.
Now let us introduce how to process signals with these new objects.

6.2 Graphon filters

The ultimate goal is to generalise the spectral filters from graphs to graphons. In order to do that we need some extra structure on our graphs. This new structure will now let us generalise our previous notions to graphons.

Definition. [Maskey et al., 2023] Let $x \in \mathbb{C}^n$ be a signal on a graph G and $\{P_1, \dots, P_n\}$ a partition of $[0,1]$ and $u \in [0,1]$. The graphon signal $\psi_x \in L^2([0,1])$ induced by x is defined as

$$\psi_x(u) := \sum_{j \leq n} x_j \chi_{P_j}(u). \tag{6.8}$$

Definition. [Maskey et al., 2023] Let $h : \mathbb{R} \to \mathbb{R}$ be a filter. Let $W \in \mathcal{W}$ be a graphon and T_W the associated GRSO (i.e. a self adjoint operator operator). Let $\{\lambda_j, \phi_j\}_j$ be the eigendecomposition of T_W. We define the realization of the filter $h(T_W)$ as

$$h(T_W)\psi = \sum_{j=1}^{\infty} h(\lambda_j)\langle \psi, \phi_j \rangle \phi_j. \tag{6.9}$$

Here $\psi \in L^2([0,1])$

As one can see these definitions are quite similar to the ones we have already seen above on spectral filters, in that sense it is almost intuitive to define graphon convolutional neural networks in the same way as spectral filters, with no other adjustments needed. This then implies that the inputs are vector valued signals. We call these signals graphon feature maps. Now we shall collect useful results in order to better understand these new objects.

Definition. [Maskey et al., 2023] For $n \in \mathbb{N}$, we define the space of step functions $\mathcal{S}_n \subset L^2([0,1])$ as $\mathcal{S}_n = span\{\chi_{P_1}, \dots, \chi_{P_n}\}$

This space is actually isomorphic to $\mathbb{C}^n$.

Lemma 6.2. *[Maskey et al., 2023] For $n \in \mathbb{N}$, the correspondence $x \to \psi_x$ defined in 6.8, between the space $\mathbb{C}^n$ and $\mathcal{S}_n$ is an isomorphism. Further, it holds*

$$\frac{1}{\sqrt{n}}\|x\|_{\mathbb{C}^n} = \|\psi_x\|_{L^2([0,1])} \quad \text{and} \quad \frac{1}{n}\langle x, y \rangle_{\mathbb{C}^n} = \langle \psi_x, \psi_y \rangle_{L^2([0,1])}$$

for all $x, y \in \mathbb{C}^n$.

Proof. A simple calculation shows the results. Let $\{P_1, \dots, P_n\}$ be a partition

of $[0,1]$ then

$$\langle \psi_x, \psi_y \rangle_{L^2([0,1])} = \int_0^1 \sum_{i=1}^n x_i \chi_{P_i}(u) \sum_{j=1}^n \overline{y_j \chi_{P_j}(u)} du \tag{6.10}$$

$$= \int_0^1 \sum_{i=1}^n \sum_{j=1}^n x_i \chi_{P_i}(u) \overline{y_j \chi_{P_j}(u)} du = \frac{1}{n} \sum_{i=1}^n x_i \overline{y_i} = \frac{1}{n} \langle x, y \rangle_{\mathbb{C}^n}. \tag{6.11}$$

The other claim is analogous. □

This lemma gives us a connection between the eigenvalues of the GSO, GWM and the graphon shift operator T_{W_A}, as shown in the next result.

Lemma 6.3. *[Maskey et al., 2023] Let G be a graph with GSO Δ and associated GWM A. Denote the eigendecomposition of Δ by $\{\lambda^i, x^i\}_{i=1}^n$. Then, T_{W_A} admits the eigendecomposition $\{\lambda^i, \sqrt{n}\psi_{x^i}\}_{i=1}^n \cup \{0, \psi_j\}_{j=n+1}^\infty$, where the eigenvectors are an orthonormal basis for $L^2([0,1])$.*

The proof of this lemma, as it is just a technical calculation, will be omitted. It shows that graphon filters do not care if we first calculate the induced graphon or apply the filter on the discrete object before inducing the graphon.

Proposition 6.4. *[Maskey et al., 2023] Let G be a graph with GSO Δ and corresponding GWM A. Let $h : \mathbb{R} \to \mathbb{R}$ be continuous with $h(0) = 0$. Then, $h(T_A) = T_{W_{nh(\Delta)}}$.*

A generalization of this proposition is given below.

Proposition 6.5. *[Maskey et al., 2023] Ler G be a graph with GSO Δ and corresponding GWM A. Let $h : \mathbb{R} \to \mathbb{R}$ be continuous. Then for every signal $x \in \mathbb{C}^n$*

$$\psi_{h(\Delta)x} = h(T_{W_A})\psi_x. \tag{6.12}$$

The graphons as defined above are not yet enough to prove transferability over all GCNNons. We still have some gaps to fill in order to characterize all GRSOs.

6.2.1 General unbounded shift operator

As one might suspect, the construction of induced graphons are not enough to represent all graphon shift operators, so we need to find objects that fill those gaps. In order to accomplish this, we consider the interval $[0,1]$ and discretize it by a grid. We do this by connecting each grid point to each of its neighbours. It is almost intuitive that this describes a graphon latent space (i.e. a space from which graphons can be sampled) and a graph. This graph together with the Laplace operator however could never define a graphon in the above sense, since

the Laplace operator is not bounded, so in order to also connect these objects with our above theory we must develop the definitions further.

Let us first define the space for these new unbounded graphon shift operators.

Definition. [Maskey et al., 2023] Let $\mathcal{L}$ be a selfadjoint operator with spectrum consisting only of eigenvalues. Denote the eigenvalues, eigenspaces and projections upon eigenspaces of $\mathcal{L}$ by $\{\lambda_j, W_j, P_j\}_{j\in\mathbb{N}}$. For each $\lambda > 0$, we define the λ'th Paley-Wiener space of $\mathcal{L}$ as

$$PW_{\mathcal{L}} = \bigoplus_{j\in\mathbb{N}} \{W_j | |\lambda_j| < \lambda\}. \tag{6.13}$$

The spectral projection $P_{\mathcal{L}}(\lambda)$ onto $PW_{\mathcal{L}}(\lambda)$ is defined by

$$P_{\mathcal{L}}(\lambda) = \sum_{|\lambda_j|<\lambda} P_j. \tag{6.14}$$

One can think of this space as the space of graphons where we forget that they have eigenvalues above and below a certain value. We shall now define what it means to be an unbounded graphon shift operator using this space.

Definition. [Maskey et al., 2023] Let $\mathcal{L}$ be a self-adjoint operator defined on a dense subset of $L^2([0,1])$, with spectrum consisting only of eigenvalues. We say that $\mathcal{L}$ is an unbounded graphon shift operator (unbounded GRSO) if it is self-adjoint and for every $\lambda > 0$, there exists a graphon $W^\lambda \in \mathcal{W}$ such that

$$\mathcal{L}P_{\mathcal{L}}(\lambda) = T_{W^\lambda}. \tag{6.15}$$

This definition is very abstract, so instead we find an easily verifiable condition whether self-adjoint operators are unbounded graphon shift operators.

Proposition 6.6. *[Maskey et al., 2023] A self-adjoint operator $\mathcal{L}$ with spectrum $\sigma(\mathcal{L}) = \{\lambda_1, \lambda_2, \ldots\}$ consisting only of eigenvalues is an unbounded GRSO if and only if the eigenvalues in each compact interval are square summable.*

The idea of the proof is that, if the eigenvalues are square summable, then we can bring it into the form of the projections, while if $\mathcal{L}$ is an unbounded shift operator, then we can calculate (6.14), which shows the square summability of eigenvalues on a compact interval. Now let us check what concept of convergence needs to be applied to unbounded graphon shift operators.

Definition. [Maskey et al., 2023] Let $(G_n)_n$ be a sequence of graphs with GSOs $(\Delta_n)_n$ and associated GWMs $(A_n)_n$. We say that (G_n) converges to the

unbounded GRSO $\mathcal{L}$ if there exists a sequence of permutations $(\pi_n)_n$ such that for every $\lambda \in \mathbb{R}$

$$||P_{\mathcal{L}}(\lambda)T_{W_{\pi_n(A_n)}}P_{\mathcal{L}}(\lambda) - \mathcal{L}P_{\mathcal{L}}(\lambda)||_{L^2([0,1])\to L^2([0,1])} \to 0 \tag{6.16}$$

for $n \to \infty$.

In general, we have an analogue to the above result that every graphon has a series of graphs that converge towards it.

Proposition 6.7. *[Maskey et al., 2023] Let $\mathcal{L}$ be an unbounded GRSO. Then, there exists a sequence of weighted graphs $(G_n)_n$ with GSOs $(\Delta_n)_n$ such that $G_n \to \mathcal{L}$.*

Let us now move to the transferability results.

7 Transferability using graphons

In this final section of the thesis, we will discuss transferability results for graphons and GCNNs using graphons.

7.1 Asymptotic transferability of graph filters

In order to prove transferability, we should first investigate if for a sequence of graphs and GSOs converging to a graphon, the associated graph filters also converge to the induced GRSO in the operator norm. We have the following lemma.

Lemma 7.1. *[Maskey et al., 2023] Let $(G_n)_n$ be a sequence of graphs with GSOs $(\lambda_n)_n$ and its associated GWMs $(A_n)_n$. Let $W \in \mathcal{W}$ satisfy $G_n \to W$. Let h be a filter. Then, there exists a sequence of permutations $(\pi_n)_n$ such that*

$$||h(T_{W_{\pi_n(A_n)}}\psi_{x_n} - h(T_W)\psi||_{L^2([0,1])} \to 0. \tag{7.1}$$

The proof of this lemma is very involved and would go beyond the scope of this thesis, so we omit the proof of the above lemma. Using this lemma, we can prove that the transferability error does indeed vanish given a big enough graph size.

Theorem 7.2. *[Maskey et al., 2023] Let $(G_n)_n$ be a sequence of graphs with GSOs $(\lambda_n)_n$ and its associated GWMs $(A_n)_n$. Let $W \in \mathcal{W}$ satisfy $G_n \to W$. Let h be a filter. consider a sequence of graph signals $(x_n)_n$ on $(G_n)_n$ and a graphon signal $\psi \in L^2([0,1])$ such that $x_n \to \psi$. Then,*

$$||\psi_{h(\Delta_n)x_n} - \psi_{h(\Delta_m)x_m}|| \to 0 \tag{7.2}$$

for $m, n \to \infty$.

Proof. [Maskey et al., 2023] By Proposition 6.4 and Lemma 7.1 we have that

$$||T_{W_{nh(\pi_n(\Delta_n))}} - h(T_W)||_{L^2([0,1])\to L^2([0,1])} \to 0 \tag{7.3}$$

for $n \to \infty$. Then by applying Proposition 6.5 and Lemma 7.1, we obtain $h(\Delta_n)x_n \to h(T_W)\psi$. Applying the triangle inequality yields the result. □

Of course such asymptotic results are often not useful in practice as it is not clear that it is actually computationally efficient to use these methods instead of simpler methods.

7.2 Non-asymptotic transferability of graph filters

Asymptotic results alone are not satisfying, so we investigate at which rate the graphon filters converge. Instead of using the usual L^p norm, we use the so called Schatten p-norm (i.e. the l^p norm of the sequence of eigenvalues). With respect to this norm, we show that the filters converge linearly. One can even give a formulation in terms of the Fourier coefficients.

Definition. [Maskey et al., 2023] Let T be a compact self-adjoint operator with eigenvalues $(\lambda_i)_{i=1}^n$, then we define the Schatten p-norm $||T||_{S_p} = ||(\lambda)_{i=1}^n||_{l^p}$

Theorem 7.3. *[Maskey et al., 2023] Let h be Lipschitz continuous with Lipschitz constant $||h||_{Lip}$ and $h(0) = 0$. Let $(G_n)_n$ be a sequence of graphs with GSOs $(\Delta_n)_n$ and associated GWMs $(A_n)_n$ such that there exists a $W \in \mathcal{W}$ with $G_n \to W$. Then, there exists a sequence of permutations $(\pi_n)_n$ such that for every $2 < p < \infty$*

$$||T_{W_{nh(\pi_n(\Delta_n))}} - T_{W_{mh(\pi_m(\Delta_m))}}||_{S_p} \leq 2||h||_{Lip}K_p||T_{W_{\pi_n(\Delta_n)}} - T_{W_{\pi_m(\Delta_m)}}||. \tag{7.4}$$

The right-hand side of this inequality converges to 0 for $m, n \to \infty$ and K_p is a universal constant given in [Potapov and Sukochev, 2011] theorem 1.

Proof. Let $(G_n)_n$ be a sequence of graphs such that $G_n \to W$. Then, by [Borgs et al., 2008] and [Janson, 2010] we know that permutations $(\pi_n)_n$ with $||T_{W_{\pi_n(A_n)}} - T_W||_{S_p} \to 0$ for $n \to 0$ exist. Every Lipschitz continuous function h with Lipschitz constant $||h||_{Lip}$ satisfies

$$||h(A) - h(B)||_{S_p} \leq ||h||_{Lip}K_p||A - B||_{S_p} \tag{7.5}$$

for self-adjoint operators A and B such that $||A - B||_{S_p} < \infty$ ([Potapov and

Sukochev, 2011]). Hence, we obtain

$$||h(T_{W_{\pi_n(A_n)}} - h(T_W)||_p \leq K_p||h||_{Lip}||T_{W_{\pi_n(A_n)}} - T_W||_p. \tag{7.6}$$

Using Proposition 6.4 and the triangle inequality finishes the proof. □

8 Generalizing to graph neural networks

In this final section, we generalise the transferability from just graph filters to GCNNs and graphon convolutional neural networks (GCNNons). In order to do that, one can not simply use the operator norm, as GCNNs and GCNNons are not linear operators. So we must introduce a new norm in which we can prove that under certain conditions we have asymptotic convergence and sometimes even linear approximation rate.

8.1 Asymptotic transferability of graph neural networks

Let us first aim to generalise Theorem 7.2 to our new setting of GCNNons. For that, we introduce a norm on $L^2([0,1])^d$ in order to know what it means for graphon feature maps to converge.

Definition. [Maskey et al., 2023] Let $d \in \mathbb{N}$ and $f \in L^2([0,1])^d$ be a graphon feature map and $x \in \mathbb{C}^{n\times d}$ a graph feature map.
We define the norms

$$||f||_{L^2([0,1])^d} = \max_{k=1,\dots,d}||f^k||_{L^2([0,1])} \tag{8.1}$$

$$||f||_{\mathbb{C}^{n\times d}} = \max_{k=1,\dots,d}||x^k||_{\mathbb{C}^n}. \tag{8.2}$$

We say that f is normalized if $\max_{k=1,\dots,d}||f^k||_{L^2([0,1])} \leq 1$. We say that x is normalized if $\max_{k=1,\dots,d}||x^k||_{\mathbb{C}^n} \leq 1$.
Let $(G_n)_n$ be a sequence of graphs with GSOs and $(x_n)_n$ a sequence of graph feature maps with $x_n \in C^{n\times d}$ for every $n \in \mathbb{N}$. Let x_n^k and f^k denote the k-th feature map of x_n and f. We say that $(x_n)_n$ converges to f if for every $k = 1, \dots, n$ it holds $x_n^k \to f^k$.

First, we show that spectral convolutional networks (SCNNs) are transferable between graphons and induced graphons. Second we use the obtained approximation to show transferability on GNNs.

Lemma 8.1. *[Maskey et al., 2023] Let $\phi := (H, M, \rho)$ be an SCNN with L layers and Lipschitz continuous activation function $\rho : \mathbb{R} \to \mathbb{R}$. Let f_l be the number of features on the l-th layer, for $l = 1 \dots, L$. Let $(G_n)_n$ be a sequence of graphs*

with GSOs $(\Delta_n)_n$ *and associated GWMs* $(A_n)_n$*. Let* $W \in \mathcal{W}$ *with* $G_n \to W$*. Suppose that* $(y_n)_n$ *is a sequence in* $(L^2([0,1]))^{f_0}$ *and* $y \in (L^2([0,1]))^{f_0}$ *such that*

$$||y_n - y||_{L^2([0,1])^{f_l}} \to 0 \tag{8.3}$$

for $n \to \infty$*. Then there exists a sequence of permutations* $(\pi_n)_n$ *such that,*

$$||\phi_{W_{\pi_n(A_n)}}(y_n) - \phi_W(y)||_{L^2([0,1])^{f_l}} \to 0 \tag{8.4}$$

for $n \to \infty$*.*

We generalise Proposition 6.5 to this setting.

Lemma 8.2. *[Maskey et al., 2023] Let* $\phi := (H, M, \rho)$ *be an SCNN. Let* G *be a graph with GSO* Δ *and associated GWM* A*. Let* $x \in \mathbb{C}^{n \times d}$ *and activation function* $\rho : \mathbb{R} \to \mathbb{R}$*. Then we have that* $\psi_{\phi_\Delta(x)} = \Delta_{W_A}(\psi_x)$*.*

The asymptotic transferability result for GCNNs now follows directly from an application of the triangle inequality.

Theorem 8.3. *[Maskey et al., 2023] In the setting of Lemma 8.1 let* $(x_n)_n$ *be a sequence of graph feature maps and* $y \in (L^2([0,1]))^{F_0}$*, such that* $x_n \to y$*. Then, there exists a sequence of permutations* $(\pi_n)_n$*, such that*

$$||\psi_{\phi_{\pi_n(\Delta_n)(x_n)}} - \psi_{\phi_{\pi_m(\Delta_m)(x_m)}}|| \to 0 \tag{8.5}$$

as $n, m \to \infty$*.*

Proof. As we have already seen in the proof of Theorem 7.2 above we can use Lemma 8.1 and Lemma 8.2 in order to obtain $(\pi_n)_n$ such that

$$||\psi_{\phi_{\pi_n(\Delta_n)(x_n)}} - \phi_W(y)||_{L^2([0,1])^{F_L}} \to 0 \tag{8.6}$$

for $n \to \infty$. This yields the result via triangle inequality. □

Next, we prove linear stability conditions. In order to achieve a rigorous proof one needs to impose some smoothness conditions on the activation functions of the GCNNs we investigate.

8.2 Non-asymptotic transferability of graph neural networks

As already mentioned above, proving transferability for all GCNNs is a daunting task, so in order to prove transferability we restrict ourselves to the class of GCNNs with a so called contractive activation function.

Definition. [Maskey et al., 2023] An activation function $\rho : \mathbb{R} \to \mathbb{R}$ is called contractive if $|\rho(x) - \rho(y)| \leq |x - y|$

In order to find linear convergence of transferability on GNNs, we assume higher regularity of the activation function and then generalise as we already did before.

Lemma 8.4. *[Maskey et al., 2023] Let $\phi := \phi(H, M, \rho)$ be an SCNN with contractive activation function ρ. Let G be a graph with GSO Δ and associated GWM A. Let $W \in \mathcal{W}$ and suppose that*

$$||A||_{\mathbb{C}^n \to \mathbb{C}^n} \leq \Gamma, \qquad ||T_W||_{L^2([0,1]) \to L^2([0,1])} \leq \Gamma \tag{8.7}$$

for $\Gamma > 0$. Let $\varepsilon > 0$ and suppose there exists a permutation π on the nodes of G with

$$||T_{W_{\pi(A)}} - T_W||_{L^2([0,1]) \to L^2([0,1])} \leq \varepsilon. \tag{8.8}$$

Suppose that for every filter h in the SCNN ϕ, we have $h \in C^1[-\Gamma, \Gamma]$, with Lipschitz derivative and $||h||_{L^\infty[-\Gamma,\Gamma]} \leq 1$. Then, there exists a $C_L \in \mathbb{N}$ such that

$$||\phi_{W_{\pi(A)}}(\psi_x) - \phi_W(y)||_{L^2([0,1])^{F_L}} \leq C_L \varepsilon. \tag{8.9}$$

Here C_L depends on M and H.

The proof of Lemma 8.4 is a long calculation and is as such omitted, but can be found in [Maskey et al., 2023]. The final result is now a direct corollary of Lemma 8.4.

Theorem 8.5. *[Maskey et al., 2023] Let $\phi := (H, M, \rho)$ be an SCNN with contractive activation function ρ. Let G_1 and G_2 be graphs with GSOs Δ_1, Δ_2 and associated GWM A_1, A_2. Suppose that*

$$||A_1||_{\mathbb{C}^n \to \mathbb{C}^n} \leq \Gamma, \qquad ||A_2||_{\mathbb{C}^n \to \mathbb{C}^n} \leq \Gamma \tag{8.10}$$

for $\Gamma > 0$. Let $\varepsilon > 0$ and suppose there exist vertex permutations π_1 of G_1 and π_2 of G_2 satisfying

$$||T_{W_{\pi_1(A_1)}} - T_{W_{\pi_2(A_2)}}||_{L^2([0,1]) \to L^2([0,1])} \leq \varepsilon. \tag{8.11}$$

Let x_1 and x_2 be normalized graph feature maps satisfying

$$||\psi_{x_1} - \psi_{x_2}||_{L^2([0,1])^{F_0}} \leq \varepsilon. \tag{8.12}$$

Then we obtain

$$||\psi_{\phi_{\pi(\Delta_1)}}(x_1) - \psi_{\phi_{\pi(\Delta_2)}}(x_2)|| \leq C_L \varepsilon, \tag{8.13}$$

where C_L is the same constant as in Lemma 8.4.

As such we have shown the linear approximation rate even holds for GCNNs.

9 Conclusion

We have learned that spectral neural networks based on graphs are transferable when using a class of functions as general as H^1 functions. Furthermore, one can expect SCNNs to be transferable when using very general objects. We can even expect the bounds to grow linearly.

However, there are some limitations of the findings for spectral methods and graphon methods. For example, determining if a function is in the Cayley smoothness space might prove difficult.

While the results for SCNNS using graphons seem very promising, we must concede that there are some strong smoothness conditions on the activation function, in general one would like to prove these results using only continuous functions.

To conclude, this thesis compiles the results of mainly two papers that use two separate approaches in order to prove transferability of GNNs. By condensing the information of these papers it was possible to find a new generalization in the form of Theorem **??**. Hopefully, this generalization can be useful in finding bounds of transferability for more general functions.

References

Christian Borgs, Jennifer T Chayes, László Lovász, Vera T Sós, and Katalin Vesztergombi. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851, 2008.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 2016.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Svante Janson. Graphons, cut norm and distance, couplings and rearrangements. *arXiv preprint arXiv:1009.2376*, 2010.

Ron Levie, Elvin Isufi, and Gitta Kutyniok. On the transferability of spectral graph filters. In *2019 13th International conference on Sampling Theory and Applications (SampTA)*, pages 1–5. IEEE, 2019.

Sohir Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: an extended graphon approach. 2022. URL https://arxiv.org/abs/2109.10096.

Sohir Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: an extended graphon approach. *Applied and Computational Harmonic Analysis*, 63:48–83, 2023.

Denis Potapov and Fedor Sukochev. Operator-lipschitz functions in schatten–von neumann classes. 2011.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

PK Suetin. Va steklov's problem in the theory of orthogonal polynomials. *Journal of Soviet Mathematics*, 12:631–682, 1979.