Convergence laws of graph neural networks on large random graphs

Candidate no. 1069907

Word count: 20769

A thesis submitted for the degree of

MSc in Advanced Computer Science

Trinity 2023

Abstract

Graph neural networks have recently surged to the forefront of computational research due to their potential and applicability in diverse domains. As their prominence grows, it becomes imperative to understand their constraints and limitations. This thesis explores an expressivity limitation, known as the zero-one law, that was only very recently discovered within particular instances of Message Passing Graph Neural Networks (MPGNNs). The goal of this thesis is to understand if this expressivity limitation is present on a larger class of MPGNNs, namely attention networks.

To this end, we introduce a novel MPGNN, generalizing the principles of graph attention mechanisms. Through both theoretical proofs and empirical tests, we demonstrate that this generalized network adheres to the zero-one law under various random graph distributions. Subsequently, we pivot our attention to a recently published paper in the domain. After a comprehensive analysis of its methodology, we extend its techniques to delve deeper into the convergence of specific attention models, and obtain zero-one laws of particular model instances. Additionally, we propose a novel experimental technique to verify the convergence behaviour.

Contents

Appendices

List of Figures

1
Introduction

As the world's digital data proliferates at an ever-increasing rate, it has become increasingly important to develop sophisticated methods of analyzing and interpreting this vast sea of information. In this context, graph data emerges as a powerful format due to its inherent ability to encapsulate multifaceted relationships and interdependencies among entities, a characteristic that is not typically represented directly in other data formats. In many disciplines and industries, from social networking [1] to biology [2], physics [3], or telecommunications [4], systems can be naturally represented as graphs, with entities as nodes and relationships as edges.

However, harnessing the power of graph data requires specialized computational techniques and this is where graph machine learning (GML) comes into play. GML, a subset of machine learning, is dedicated to developing algorithms that can analyze and learn from graph-structured data. By leveraging the unique properties of graph data, these algorithms can make predictions or draw insights that would be otherwise very difficult. A popular class of models in GML are Graph Neural Networks [5, 6].

There are different types of problems one can tackle using graph ML [7, 8]. Node-level tasks involve making predictions or inferences about individual nodes within a graph. The objective is to learn a function that can map nodes to a set of labels or continuous values. Examples of such tasks include *node classification*, *node regression*, and *node attribute prediction*. In the context of social networks, a node-level task could be predicting the political affiliation of a user (node) based on their connections and activities. In bioinformatics, a similar task might be predicting the functions of a protein in a protein-protein interaction network [9–11].

Edge-level tasks, as the name suggests, focus on the edges or connections in the graph. The aim here is to learn a function that can predict or infer the properties of the edges. Common tasks include link prediction, link classification, and relationship prediction. For example, in a citation network, link prediction could involve predicting future collaborations between researchers [12].

In a recommender system, users can be represented as nodes and their preferences or "liking" for a movie can be depicted as edges (links) between them. The system then tries to predict the likelihood of an edge forming between a user node and a movie node, thus providing the user with a recommendation [13].

Lastly, graph-level tasks involve making predictions for entire graphs. Here, the aim is to learn a function that maps an entire graph to a target label or a continuous value. These tasks include *graph classification*, *graph regression*, and *graph generation*. An example of a graph-level task is classifying molecules for drug discovery, where each molecule can be represented as a graph, and the task is to predict its properties, such as the toxicity number or efficacy [14–16].

While the applications of graph machine learning are vast and the potential of graph data remains undeniably powerful, it is imperative to understand and address the limitations of these methods. The purpose of this thesis is to study a limitation of *message passing graph neural networks* (MPGNNs), which is a framework encompassing many popular graph neural network architectures used in practice.

1.1 Problem statement and outline of the thesis

This thesis is about exploring a novel limitation of MPGNNs, that has only recently come to light. This limitation, which we will further elaborate upon in due course, has thus far been studied only within a few MPGNNs models. A primary aim of this thesis is to broaden our understanding of this limitation by examining its potential presence in a larger class of models, particularly attention models. Our investigation is largely theoretical, and relies on probabilistic techniques. We will design experiments that test whether our theoretical predictions are observed in actual models.

The thesis can be summarized as follows:

- We begin by motivating and introducing the recent discovery of a new limitation of MPGNNs - the so called zero-one law [17]. This is done in Chapter 3.

- In Chapter 4, we introduce attention models and begin our study of the zero-one law on these models, which was left open in previous work. We will define a special type of attention models, and prove and experimentally verify the zero-one law. In Chapter 5, we extend our findings to different settings

- In Chapter 6, we explore the zero-one law for other attention models. Using a recent paper in this field [18], we develop a theoretical framework capturing a large class of models. After doing so, we will extend the results of the paper, and theoretically prove the zero-one law for several attention models. Additionally, we design a novel experimental method to verify our theoretical claims.

- Finally, we discuss the limitations of our work, and propose directions for further study.

2
Background

2.1 Message passing graph neural networks

A crucial class of models that stands out in graph ML are Message Passing Graph Neural Networks (MPGNNs) [19]. They operate on the principle of passing and aggregating "messages" from the local neighborhood of nodes. This process of information exchange allows MPGNNs to learn the complex relationships between nodes in a graph and is typically performed in iterative rounds, called layers. Specifically, we consider simple, undirected, unweighted graphs $G = (V, E, \mathbf{F})$, where V is a set of vertices, E is a set of edges in the graph, and $\mathbf{F}$ is a matrix containing the *features* of each node. Typically, the features are stored as rows in $\mathbf{F}$, which means $\mathbf{F}$ is a matrix of dimensions $n \times d$, where n is the number of vertices and d is the dimension of the node features.

In the message-passing phase, every node in the graph sends a message to its neighbors, and each node receives messages from its neighbors. The content of the message is computed based on the features of the sending node, the receiving node, and possibly the edge connecting them. The most common practice is for each node to gather the messages sent by its neighbors and aggregate them into a single message. After aggregation, an update function is used to combine the aggregated message with the node's current features to produce the node's updated features. This process of message passing and aggregation is usually repeated for several layers. This can be mathematically expressed as follows. For every node $u \in V$ and $l \geq 1$, the l-th layer computes

$$\begin{aligned}
\mathbf{h}_u^0 &= \mathbf{x}_u && \text{initialize} \\
\mathbf{m}_u^l &= f^l(\mathbf{h}_u^{l-1}, \{\{\mathbf{h}_v^{l-1} | v \in N(u)\}\}) && \text{aggregate} \\
\mathbf{h}_u^l &= \phi^l(\mathbf{h}_u^{l-1}, \mathbf{m}_u^l) && \text{update/combine}
\end{aligned}$$

where $\mathbf{x}_u$ are the initial features (stored as rows in the matrix $\mathbf{F}$) of dimension d, and $N(u)$ denotes the neighborhood of the node u. The aggregation functions f^l can be a summation, average, maximum, or more sophisticated operations. The key requirement is that the aggregation function must be permutation invariant, meaning the order in which messages are received does not impact the aggregated result. The function ϕ^l can be as simple as an addition operation or a non-linearity. Note that $\mathbf{h}_u^l$ can be of a different dimension as the initial node features, and we will denote this dimension as d_l, and write $\mathbf{h}_u^l \in \mathbb{R}^{d_l}$, with $d_0 = d$.

Different choices of the aggregation and update functions yield different models, suited for different tasks. We will now list some common MPGNNs that we will refer to throughout the document, but we first need some auxiliary definitions:

Definition 2.1 (ReLU).
The ReLU is a function from $\mathbb{R}$ to $\mathbb{R}$ defined as

$$ReLU(t) = \begin{cases} t, & \textit{if } t \geq 0 \\ 0, & \textit{if } t < 0 \end{cases}$$

Definition 2.2 (LeakyReLU).
The LeakyReLU is a function from $\mathbb{R}$ to $\mathbb{R}$ with parameter $0 < k < 1$ defined as

$$LeakyReLU(t) = \begin{cases} t, & \textit{if } t \geq 0 \\ kt, & \textit{if } t < 0 \end{cases}$$

It is common to apply the ReLU and LeakyReLU to vectors. In this case, the functions are applied to each component of the input vector.

Definition 2.3 (Softmax).
The softmax function is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$ for $d \in \mathbb{N}$, defined as

$$softmax(x_1, x_2 \dots, x_d) = \left(\frac{\exp(x_1)}{\sum_{k=1}^d \exp(x_k)}, \frac{\exp(x_2)}{\sum_{k=1}^d \exp(x_k)}, \dots, \frac{\exp(x_d)}{\sum_{k=1}^d \exp(x_k)} \right)$$

We denote the j-th component as $softmax_j(\mathbf{x})$

We are now ready to define some models. A very famous architecture is the *graph convolutional network* (GCN), introduced by Kipf and Welling [20].

Definition 2.4 (GCN).
A graph convolutional network (GCN) is a MPGNN which aggregates as $\mathbf{x}_v^l = \sigma(\mathbf{y}_v^l)$, where σ is a non-linearity of choice (such as ReLU), and $\mathbf{y}_v^t$

$$\mathbf{y}_v^l = \sum_{u \in N^+(v)} \frac{1}{\sqrt{|N(u)||N(v)|}} \psi^l(\mathbf{x}_u^{l-1}) + \mathbf{b}^l$$

where $\psi^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ is a linear map (matrix) and $N^+(v)$ denotes the extended neighborhood of the node v defined as $N^+(v) = \{v\} \cup \{u \in V | (u, v) \in E\}$

Definition 2.5 (MeanGNN).
A GNN with mean aggregation and global readout (MeanGNN) is an MPGNN with the aggregation scheme given by $\mathbf{x}_v^l = \sigma(\mathbf{y}_v^l)$, where σ is a non-linearity of choice (such as ReLU), and $\mathbf{y}_v^l$

$$\mathbf{y}_v^l = \sum_{u \in N^+(v)} \frac{1}{|N^+(v)|} \psi_1^l(\mathbf{x}_u^{l-1}) + \sum_{u \in V(G)} \frac{1}{n} \psi_2^l(\mathbf{x}_u^{l-1}) + \mathbf{b}^l$$

where $\psi_1^l, \psi_2^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ are linear maps (matrices).

Definition 2.6 (SumGNN).
A GNN with sum aggregation and global readout (SumGNN) is a MPGNN with the aggregation scheme given by $\mathbf{x}_v^l = \sigma(\mathbf{y}_v^l)$, where σ is a non-linearity of choice (such as ReLU), and $\mathbf{y}_v^l$

$$\mathbf{y}_v^l = \psi_1^l(\mathbf{x}_v^{l-1}) + \sum_{u \in N^+(v)} \psi_2^l(\mathbf{x}_u^{l-1}) + \sum_{u \in V(G)} \psi_3^l(\mathbf{x}_u^{l-1}) + \mathbf{b}^l$$

where $\psi_1^l, \psi_2^l, \psi_3^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ are linear maps (matrices).

After the message passing phase, it can be useful to combine the final features into a single vector. This is done by using a *pooling layer.* We will often use a mean-pooling:

Definition 2.7 (Mean-pooling).
A mean-pooling layer is a function which takes as input an arbitrary number of vectors $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n$ *of the same dimension, and outputs their mean* $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$.

An important class of MPGNNs are models which aggregate using attention. These models will be the focus of this thesis, and will be explained in more detail later. Right now we will just list the definitions.

Definition 2.8 (Single-head transformer).
We call an MPGNN a single-head transformer, if it satisfies the aggregation scheme given by $\mathbf{x}_i^l = \sigma(\mathbf{y}_i^l)$, *where* σ *is a non-linearity of choice (such as ReLU), and*

$$\mathbf{y}_i^l = \sum_{j \in N^+(i)}^{n} softmax_j \left(\frac{\langle \psi_1^l(\mathbf{x}_i^{l-1}), \psi_2^l(\mathbf{x}_j^{l-1} \rangle)}{\sqrt{d^l}} \right) \psi^l(\mathbf{x}_j^{l-1})$$

where $\psi_1^l, \psi_2^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ *are linear maps (matrices), and* $\langle \cdot, \cdot \rangle$ *denotes the dot product.*

Definition 2.9 (Single-head graph attention network).
We call an MPGNN a single-head graph attention network (GAT), if it satisfies aggregation scheme given by $\mathbf{x}_i^l = \sigma(\mathbf{y}_i^l)$, *where* σ *is a non-linearity of choice (such as ReLU), and*

$$\mathbf{y}_i^l = \sum_{j \in N^+(i)}^{n} softmax_j(LeakyReLU(\langle \mathbf{a}^l, \psi^l(\mathbf{x}_i^{l-1}) || \psi^l(\mathbf{x}_j^{l-1}) \rangle)))\psi^l(\mathbf{x}_j^{l-1})$$

where $||$ *denotes concatenation of vectors,* $\psi^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ *is a linear map (matrix) and* $\mathbf{a}^l \in \mathbb{R}^{2d_l}$.

Definition 2.10 (Single-head GATv2).
We call an MPGNN a single-head GATv2, if it satisfies the aggregation scheme given by $\mathbf{x}_i^l = \sigma(\mathbf{y}_i^l)$, *where* σ *is a non-linearity of choice (such as ReLU), and*

$$\mathbf{y}_i^l = \sum_{j \in N^+(i)}^{n} softmax_j(\langle \mathbf{a}^l, LeakyReLU(\psi^l(\mathbf{x}_i^{l-1}) || \psi^l(\mathbf{x}_i^{l-1}))\rangle)\psi^l(\mathbf{x}_j^{l-1})$$

where $||$ denotes concatenation of vectors, $\psi^l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ is a linear map (matrix) and $\mathbf{a}^l \in \mathbb{R}^{2d_l}$.

These attention models are simplifications of actual models. For example, Definition 2.8, is inspired by the famous transformer model proposed in [21]. However, real transformers use multiple attention heads instead of just one, and the same applies to GATs and GATv2s, which are inspired by [22] and [23] respectively.

MPGNNs have been a significant advancement in GNNs due to their flexibility and generality. By designing different message, aggregation, update, and readout functions, a wide range of GNN models can be represented in the MPNN framework.

2.2 Sub-Gaussian random variables

In this section, we will provide an overview of sub-Gaussian random variables, based on [24, 25]. The definitions and theorems below will be crucial in later chapters.

Sub-Gaussian random variables are variables which are quite tightly concentrated around their mean. More precisely, they are variables with tails which decay quickly, with the rate of the decay controlled by the Gaussian distribution. This is why they are called sub-Gaussian. We have the following definition

Definition 2.11

A random variable X with $\mathbb{E}[X] = 0$ is called sub-Gaussian if there is a constant $\sigma > 0$ such that for any $t > 0$

$$\max\{P(X \geq t), P(X \leq -t)\} \leq \exp\left(\frac{-t^2}{2\sigma^2}\right)$$

The constant σ^2 is often called the variance proxy. We write $X \sim subG(\sigma^2)$.

To prove a variable is sub-Gaussian, one can instead consider the following condition:

Theorem 2.1

Let X be a random variable with $\mathbb{E}[X] = 0$, such that for any $s \in \mathbb{R}$, we have

$$\mathbb{E}[\exp(sX)] \leq \exp\left(\frac{\sigma^2 s^2}{2}\right)$$

Then, $X \sim subG(\sigma^2)$

The proof follows directly from Definition 2.11, after applying Markov's inequality.

Theorem 2.1 gives us an alternative way to prove a vector is sub-Gaussian, but it is natural to ask whether the implication holds also in the other direction. The answer is yes, but only up to a multiplicative constant.

Theorem 2.2

Let $X \sim subG(\sigma^2)$. Then for any $s > 0$, we have

$$\mathbb{E}[\exp(sX)] \leq \mathbb{E}[\exp\left(\frac{24\sigma^2 s^2}{2}\right)]$$

The proof is much more involved.

Theorem 2.1 and Theorem 2.2 give us an equivalent definition of a vector being sub-Gaussian, up to a multiplicative constant. However, for all our purposes, we do not care about this multiplicative constant. Thus we will ignore the constant 24 and use instead

$$X \sim subG(\sigma^2) \iff \mathbb{E}[\exp(sX)] \leq \mathbb{E}[\exp\left(\frac{\sigma^2 s^2}{2}\right)]$$

This is without loss of generality since all results can trivially be extended to take into account this constant.

We can now define sub-Gaussian random *vectors*. This is important for our purposes.

Definition 2.12

A random vector $\mathbf{x} \in \mathbb{R}^d$ is said to be sub-Gaussian with variance proxy σ^2, if $\mathbb{E}[X] = 0$ and for any $\mathbf{u} \in \mathbb{R}^d$ such that $\|\mathbf{u}\|_2 = 1$, the scalar random variable $\langle \mathbf{u}, \mathbf{x} \rangle$ is sub-Gaussian with variance proxy σ^2. We write $\mathbf{x} \sim subG_d(\sigma^2)$.

Crucially, one has the following consequence.

Lemma 2.3

Let $\mathbf{x} \sim subG_d(\sigma^2)$. Then any component of $\mathbf{x}$ is $subG(\sigma^2)$.

Proof. Let $\mathbf{e}_i$ denote the ith standard basis vector. Then $x_i = \langle \mathbf{e}_i, \mathbf{x} \rangle \sim subG(\sigma^2)$ by definition. □

The implication holds in the other direction, assuming independence.

Lemma 2.4

Let $X_1, \ldots, X_d$ be independent $subG(\sigma^2)$ real random variables. Then the random vector $\mathbf{x} = (X_1, \ldots, X_d) \in \mathbb{R}^d$, is $subG_d(\sigma^2)$.

Proof. Let u be an arbitrary vector in $\mathbb{R}^d$, with $\|\mathbf{u}\|_2 = 1$. We have, for any $s \in \mathbb{R}$

$$\begin{aligned}\mathbb{E}[\exp(s\langle \mathbf{u}, \mathbf{x}\rangle)] = \prod_{i=1}^{d} \mathbb{E}[\exp(su_i x_i)] \leq \prod_{i=1}^{d} \exp(\frac{\sigma^2 (su_i)^2}{2}) \\ = \exp(\frac{\sigma^2 s^2 \|\mathbf{u}\|_2}{2}) = \exp(\frac{\sigma^2 s^2}{2})\end{aligned}$$

where we used independence in the first step, and Theorem 2.2 in the second step. Now apply Theorem 2.1 to conclude that $\langle \mathbf{u}, \mathbf{x}\rangle \sim subG(\sigma^2)$. □

Given these results, we can formulate a lemma which will be very important later:

Lemma 2.5

Let $X_1, \ldots X_n$ be independent $subG(\sigma^2)$ random variables. Then for any vector $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_n)$, it holds that

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i| > t) \leq 2 \exp\left(\frac{-t^2}{\sigma^2 ||a||_2^2}\right)$$

Proof. Let $\mathbf{x} = (X_1, \ldots X_n)$. By Lemma 2.4, we know $\mathbf{x} \sim subG_n(\sigma^2)$. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_n)$ be arbitrary. If we define $\mathbf{y} = \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|_2}$, then $\mathbf{y}$ is a unit vector. By Definition 2.11, we have $\langle \mathbf{x}, \boldsymbol{\alpha}\rangle \sim subG(\sigma^2)$, and thus

$$\mathbb{P}(|\langle \mathbf{x}, \boldsymbol{\alpha}\rangle| \geq t) = \mathbb{P}(|\langle \mathbf{x}, \mathbf{y}\rangle \geq \frac{t}{\|\boldsymbol{\alpha}\|_2}) \leq 2 \exp\left(-\frac{t^2}{\sigma^2 \|\boldsymbol{\alpha}\|_2}\right)$$

by Definition 2.11. □

2.3 A primer on probability and analysis

This section contains some standard results from mathematical analysis and probability theory that will be very useful in later sections.

Theorem 2.6 (Chebyshev's inequality).

For every square-integrable random variable X and every $a > 0$

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq a) \leq \frac{Var[X]}{a^2}$$

Definition 2.13 (Almost sure convergence).

*Let (X_n) be a sequence of random variables defined on the probability space $(\Omega, \Sigma, \mathbb{P})$. We say X_n converges **almost surely** (a.s) to a random variable X defined on the same probability space, if*

$$\mathbb{P}\left(\{\omega \in \Omega | \lim_{n\to\infty} X_n(\omega) = X(\omega)\}\right) = 1$$

Definition 2.14 (Convergence in distribution).

*Let (X_n) be a sequence of random variables with cumulative distribution functions $F_1, F_2 \dots$. We say X_n converges **in distribution** to a random variable X with cumulative distribution function F, if $F_n(x) \to F(x)$ for all x where F is continuous.*

The concepts of almost sure convergence and convergence in distribution are related. In fact, the following lemma implies that almost sure convergence is in fact stronger than convergence in distribution.

Lemma 2.7

Let $(X_n) \to X$ a.s. Then $(X_n) \to X$ in distribution.

The converse does not hold.

Theorem 2.8 (Strong law of large numbers).

Let (X_n) be a sequence of i.i.d random variables, with $\mathbb{E}[X_1] = \mu$. Then

$$\frac{X_1 + X_2 + \dots X_n}{n} \to \mu \ a.s$$

In view of Lemma 2.7 we could formulate the above with convergence in probability. This is called the weak law of large numbers.

Theorem 2.9 (Continuous mapping theorem).

Let (X_n) be a sequence of random variables, and assume $X_n \to X$ in distribution. Let C be a function and let D_C denote the set of discontinuities of C. Assume that $\mathbb{P}(X \in D_C) = 0$. Then $C(X_n) \to C(X)$ in distribution.

We will also need some concentration inequalities. The first is a bound on binomial random variables:

Theorem 2.10 (A Chernoff Bound).
Let $X_1, X_2, \ldots, X_n$ be independent Ber(p_i) random variables. Let $X = \sum_{i=1}^{N} X_1$ and $\mu = \mathbb{E}[X] = \sum_{i=1}^{N} p_i$. Then for $0 < \delta < 1$

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp(-\mu\delta^2/3)$$
$$\mathbb{P}(X \leq (1-\delta)\mu) \leq \exp(-\mu\delta^2/2)$$

Theorem 2.10 is one version of Chernoff bounds. In particular, one can obtain a tighter inequality, but the above is sufficient for our purposes.

The next concentration inequality is a remarkable theorem, because of its generality and power. This theorem will be quite crucial in later sections.

Theorem 2.11 (Multidimensional McDiarmid's inequality).
Let $g : S^{n-1} \to \mathbb{R}^d$ be a function of n variables. Suppose that g satisfies a vectorial version of the bounded differences property, meaning that whenever x and x' differ in only the ith component, we have

$$\|g(\mathbf{x}) - g(\mathbf{x}')\|_\infty \leq c_i$$

then for any independent random variables $X_1, ..., X_n$ it holds that

$$\|g(X_1, \ldots X_n) - \mathbb{E}[g(X_1, \ldots X_n)]\|_\infty \leq \sqrt{\frac{1}{2}\sum_{i=1}^{n-1} c_i^2 \ln(\frac{2d}{\rho})}$$

holds with probability at least $1 - \rho$.

Theorem 2.12 (Dominated Convergence Theorem).
Let (X, Σ, μ) be a measure space. Let $u : X \to \mathbb{R}$ be Σ-measurable and suppose $\int_X u dP < \infty$. Let $f_j : X \to \mathbb{R}$ be a sequence of Σ measurable functions with $|f_j| \leq u$ $\forall j$. Suppose $f_j \to f$ almost everywhere, and assume f is Σ-measurable. Then

$$\lim_{j \to \infty} \int_X |f - f_j| d\mu = 0$$

This theorem can be extended to the case of vector functions, which is what we will actually need.

Theorem 2.13 (Dominated Convergence Theorem for random vectors).
Let $\mathbf{X}_n$ be a sequence of random vectors in $\mathbb{R}^d$, with $\mathbf{X}_n \to \mathbf{X}$ a.s. Suppose $\|\mathbf{X}_n\|_\infty \leq Y$ for some random variable Y with $\mathbb{E}[Y] < \infty$. Then

$$\lim_{n\to\infty} \|\mathbb{E}(\mathbf{X}_n) - \mathbb{E}(\mathbf{X})\|_\infty = 0$$

Proof. For every component $i \in \{1, \dots d\}$, we have $|(\mathbf{X}_n)_i| \leq Y$ and $(\mathbf{X}_n)_i \to (\mathbf{X})_i$. Thus we can apply 2.12 to every component, to get that $\mathbb{E}[(\mathbf{X}_n)_i] \to \mathbb{E}[(\mathbf{X})_i]$, and the result follows. □

We will also need some basic results about norms of vector spaces and linear operators.

Theorem 2.14 (Cauchy-Schwarz Inequality).
Let $(V, \langle \cdot, \cdot \rangle)$ be an inner product space and $\|\cdot\|$ be the induced norm. Then for any $\mathbf{x}, \mathbf{y} \in V$, we have

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|$$

Lemma 2.15 (Norm equivalence).
Let V be finite-dimensional and let $\|\cdot\|, \|\cdot\|_*$ be two norms on V. Then they are equivalent, meaning $\exists C_1, C_2 > 0$ such that $\forall \mathbf{x} \in V$, we have

$$C_1 \|\mathbf{x}\| \leq \|\mathbf{x}\|_* \leq C_2 \|\mathbf{x}\|$$

Lemma 2.16 (Boundedness of linear operators).
Let V be finite-dimensional and let $\psi : V \to W$ be a linear map. Let $\|\cdot\|_V$ be a norm on V and $\|\cdot\|_W$ be a norm on W. Define

$$\|\psi\| = \sup_{\|\mathbf{x}\|_V = 1} \|\psi(\mathbf{x})\|_W.$$

Then $\|\psi\| < \infty$.

In particular, note that for $d \in \mathbb{N}$ the vector space $\mathbb{R}^d$ is d-dimensional. Finally, we will need a few trivial statements

Lemma 2.17
Let $\mathbf{x} \in \mathbb{R}^d$ be random vector in and let $\psi : \mathbb{R}^d \to \mathbb{R}^m$ be a linear map. Then $\mathbb{E}[\psi(\mathbf{x})] = \psi(\mathbb{E}[\mathbf{x}])$.

Definition 2.15 (Lipschitz continutity).
Let function $f : \mathbb{R} \to \mathbb{R}$ and let $c \geq 0$. We say f is c-Lipschitz continuous, if $\forall x, y \in \mathbb{R}$ we have $|f(x) - f(y)| \leq c|x - y|$.

3
A review of the limitations of MPGNNs

Message Passing Graph Neural Networks (MPGNNs) have drawn significant attention due to their capability to learn graph representations and have been applied to various graph-based tasks. However, it turns out that MPGNNs suffer from some fundamental limitations which we will now discuss.

3.1 Expressive power

One of the theoretical limitations of GNNs is associated with their expressiveness, especially with respect to the 1-Weisfeiler-Lehman (1-WL) graph isomorphism test. The WL test is an iterative method for determining whether two graphs are isomorphic (i.e., identical up to a renaming of nodes), and it is defined as follows:

1. **Input:** Graph $G = (V, E)$ and an initial labeling $\lambda^0 : V \to C$ of the vertices, where C is the set of initial labels.

2. **Initialization:** All nodes $u \in V$ are initialized to their initial labels $\lambda^0(u)$.

3. **Refinement:** All nodes $u \in V$ are recursively re-labeled:

$$\lambda^{i+1}(G, u) = \tau \left(\lambda^i(G, u), \{\{\lambda^i(G, v) | v \in N(u)\}\} \right)$$

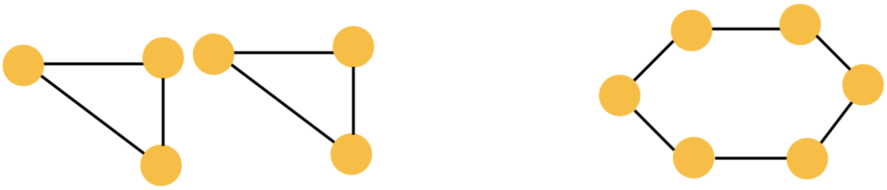


Figure 3.1: Two non-isomorphic graphs

where $\{\{\cdot\}\}$ denotes a multiset (a set with repetitions) and τ is a bijection that maps a pair (l, S), where c is a label and S a multiset of labels, to a unique label.

4. **Stop**: The algorithm terminates at the first iteration j where

$$\forall u, v \in V : \lambda^{j+1}(G, u) = \lambda^{j+1}(G, v) \iff \lambda^{j}(G, u) = \lambda^{j}(G, v)$$

In each iteration, every node is assigned a new label based on its current label and the multiset of labels of its neighbors. The algorithm terminates once the labelling becomes 'stable'. If at any iteration two graphs have different label distributions, they are deemed non-isomorphic.

Crucially, for some types of graphs, the 1-WL test fails to detect that they are in fact not isomorphic. In Figure 3.1, we can see two graphs that are not isomorphic, but the 1-WL test cannot distinguish between them. It is a famous result [26, 27], that GNNs are, in general, at most as expressive as the 1-WL test. This means that there are non-isomorphic graphs that cannot be distinguished by GNNs, and as a result, GNNs might classify non-isomorphic graphs identically, regardless of the target function to be captured.

To address the limitations of GNNs in the context of the 1-WL test, researchers have explored the concept of higher-order GNNs [27]. These models aim to capture more complex graph structures by aggregating information beyond immediate neighborhoods, essentially trying to emulate the behavior of k-dimensional versions of the WL test. However, a significant challenge with higher-order GNNs is their computational cost. This makes it infeasible for many real-world applications, especially those involving large graphs or requiring real-time processing. Consequently, while higher-order GNNs present a promising direction to overcome the 1-WL limitation, their practical adoption remains constrained.

Another approach to tackle the expressivity constraints imposed by 1-WL, is to employ *random node initialization* (RNI). By randomly setting the initial node features, the MPGNN can more effectively detect structures, thereby boosting its expressive capabilities beyond that of 1-WL [28]. In fact, it has been shown that MPGNNs with RNI are *universal* [29]. To understand what this means, we need a definition.

Definition 3.1 ([29]).
Let $\mathcal{G}_n$ be the class of all n-vertex graphs, i.e., graphs that consist of at most n vertices, and let $f : \mathcal{G}_n \to \mathbb{R}$. We say that a randomized function X that associates with every graph $G \in \mathcal{G}_n$ a random variable $X(G)$ is an (ϵ, δ)-approximation of f if for all $G \in \mathcal{G}_n$ it holds that $\mathbb{P}(|f(G) - X(G)| \leq \epsilon) \geq 1 - \delta$

Now the universality of MPGNNs with RNI is summarized in the following theorem

Theorem 3.1 ([29]).
Let $n \geq 1$, and let $f : \mathcal{G}_n \to \mathbb{R}$ be invariant. Then, for all $\epsilon, \delta > 0$, there is an MPGNN with RNI that (ϵ, δ)-approximates f.

Because of these insights, MPGNNs with RNI are drawing a lot of interest in GML research and they are a central topic of this thesis.

3.2 Oversmoothing and oversquashing

Two prevalent challenges that arise in the application of GNNs are the problems of oversmoothing [30], and oversquashing [31]. Oversmoothing, is a phenomenon where the node features become increasingly similar to each other as the depth of the network (i.e., the number of message passing layers) increases. The more the information is propagated through the network, the more homogenized it becomes, potentially leading to indistinguishable node embeddings. This poses a problem for deep GNNs, as it can lead to a loss of node-specific information and limit the discriminative power of the network.

Oversquashing refers to the tendency of GNNs to squash information when aggregating messages from neighbors. This effect arises because of aggregation functions such as summation or average, but also the topology of the graph [32]. To explain the phenomenon, we need the following definition

Definition 3.2 (Receptive field).
Let G be a graph and let $N_i(u)$ denote the i-hop neighborhood, i.e., the set of nodes reachable from a given node via a shortest path of length i. Then the receptive field of a node u up to distance k is $N_1(u) \cup N_2(u) \ldots \cup N_k(u)$

Now consider a scenario in which we need to send message over nodes which are 'far away', say, at a distance k. To do this, we must include at least k layers, but by doing so, every node receives information from its entire receptive field up to distance k, and this can become very large — in some graphs the size of the receptive field can grow exponentially with k. However, we are still forcing the information to fit into a vector of fixed length, which means the messages are 'squashed' together, and information is lost. Oversquashing can result in loss of valuable information during the message-passing phase, and thus diminish the model's ability to capture complex patterns in the graph.

Both oversmoothing and oversquashing are major hurdles that need to be tackled for the effective application of GNNs. Naturally, these two limitations are of great interest to researchers and a significant amount of work has been conducted to understand and alleviate them. The interested reader is referred to [32–34] for the analysis of these phenomena.

3.3 Zero-one laws as a new measure of expressiveness

So far we have described a few well-known limitations of MPGNNs, and in particular, we discussed that their expressive power is upper bounded by the 1-WL test. Expressivity is a very popular topic in Graph ML research, and there have been several other works tackling various aspects of this question [35–37]. This chapter introduces and examines an expressiveness limitation that was discovered only very recently [17], and has significant implications in this area of research. The limitation is the so-called 'zero-one law' of graph neural networks, and it is the central topic of this thesis. Similarly to the 1-WL result, it gives an upper bound on the expressive power of GNNs. In particular, it tackles the question: What class of functions on graphs can be captured by a *single* GNN with random node features? Observe that this is different from Theorem 3.1, because Theorem 3.1 is not a *uniform* result — the construction of the GNN depends on the graph size [17, 29].

However, before we explain this result in greater detail, let us provide the reader with some motivation. The context in which this limitation was discovered is very interesting and is worth understanding.

3.3.1 Zero-one laws in mathematics

Zero-one laws are theorems in mathematics which say that certain classes of events occur either with probability 0 or with probability 1. Such theorems are of great interest to mathematicians, because they uncover a fundamental structure in the studied class of problems, one which would otherwise be non-trivial to observe. Let us mention two famous zero-one laws. The first zero-one law is the well-known Kolmogorov's zero-one law:

Theorem 3.2 (Kolmogorov's zero-one law).
Let (X_n) be a sequence of independent random variables, and let τ be their tail σ-algebra. Then if $E \in \tau$, then $\mathbb{P}(E) = 1$ or 0.

Given a sequence of independent random variables, denoted as X_n, from any random distribution, suppose we are interested in the extreme values of $|X_n|/n$. It is relatively straightforward to demonstrate that the *event*

$$\limsup_{n\to\infty} |X_n|/n = 0$$

is part of the so-called tail σ-algebra, leading to the conclusion that

$$\mathbb{P}(\limsup_{n\to\infty} |X_n|/n = 0) = 0 \text{ or } 1$$

This result is rather significant as it does not require any particular knowledge about the sequence X_n and demonstrating that something is part of the tail σ-algebra is usually much easier than using other methods to understand the problem. However, it is crucial to understand that the theorem does not clarify which of these two outcomes will occur, but it informs us that no other outcomes are possible. This will also be true for the other zero-one laws we will consider. The above example is very simple, but one can imagine applying this theorem in much more complicated situations, where its power becomes apparent.

A second zero-one law occurs in the realm of first-order logic, and it is the primary motivation for the zero-one laws of graph neural networks. The result concerns sentences in first order logic, evaluated on random graphs. We first need a definition.

Definition 3.3 (Erdős-Rényi random graph model).
The Erdős-Rényi random graph model (E-R model) is a process for generating random graphs. Given parameters $n \in \mathbb{N}$ and $p \in [0, 1]$, the process generates a graph G with n vertices where each edge is present with probability p, independently of other edges. To denote an E-R graph G, we write $G \sim \mathbb{G}(n, p)$.

We consider *simple* graphs (i.e., no self-loops) sampled from the Erdős-Rényi model $\mathbb{G}(n, 1/2)$, and we will evaluate logical formulas on these graphs. In particular, we consider the set of logical sentences in first-order logic of (undirected) graphs with equality. We will denote this class by Ψ. For example, the sentence "does the graph contain a triangle" is a valid sentence in Ψ, because it can be expressed as

$$\exists x, y, z \; E(x, y) \wedge E(y, z) \wedge E(z, x)$$

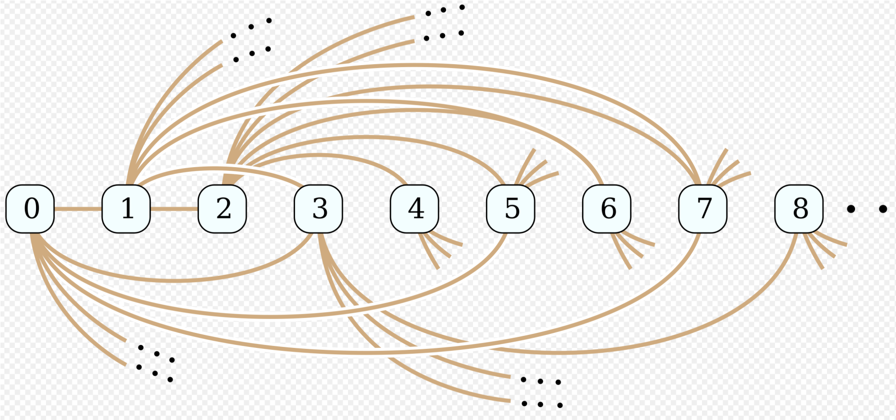


Figure 3.2: The Rado graph

where x, y, z are variables interpreted as nodes, and E is relation where $E(x, y)$ denotes and undirected edge between the interpretations of x and y.

Note that any fixed sentence in $\psi \in \Psi$ is either true or it is false on any fixed graph, and thus for any $n \in \mathbb{N}$, we can consider the probability $\mathbb{P}_n(\psi)$ of ψ being true. In the case of $p = 1/2$, all graphs are sampled with equal probability, and thus:

$$\mathbb{P}_n(\psi) = \frac{\text{\# graphs in } \mathbb{G}(n, 1/2) \text{ where } \psi \text{ is satisfied}}{2^{\binom{n}{2}}}$$

where $2^{\binom{n}{2}}$ is the number of graphs on n vertices. The key result is then summarized in the following theorem

Theorem 3.3 (Zero-one law of first order logic [38]).
For any $\psi \in \Psi$, we have $\lim_{n\to\infty} \mathbb{P}_n(\psi) = 0$ or 1.

Theorem 3.3 is a remarkable result, and we will present a (very) high-level proof of this theorem, inspired by [38–40].

Definition 3.4
The Rado graph is a countably infinite graph with vertex set $V = \mathbb{N} \cup \{0\}$ and edge set defined as follows: For any $x, y \in V$, with $x < y$ there is an edge connecting x and y iff the $x - th$ bit in the binary representation of y is 1.

For example, the vertex 0 is connected to all odd vertices, because their $0-$th bit is 1. The vertex 1 is connected to 0 and vertices with binary representation 10 or 11, which are precisely those congruent to 2 or 3 modulo 4. See Figure 3.2.

There are other (equivalent) definitions of the Rado graph. It can be defined as the limit of the Erdős-Rényi process. If one considers the set $0 \cup \mathbb{N}$, and writes an edge between any pair with probability $p \in (0,1)$, the resulting graph is isomorphic to the Rado graph with probability 1. For this reason, it is often called *the* random graph [41].

Lemma 3.4

Let $U, V \subset \mathbb{N}$ be finite with $U \cap V = \emptyset$. Then $\exists x \in \mathbb{N}$ with $x \notin U$ and $x \notin V$, such that x is adjacent to every node in U, but x is not adjacent to any node in V.

Proof. We can easily construct such an x. Firsly, we want x to be larger than the largest element of $U \cup V$ - this ensures we only need to consider the bits of x. Now, we simply set, for every $u \in U$, the $u - th$ bit of x to be 1, which will ensure there is an edge between x and u. This is obtained if we let

$$x = 2^{1+\max(U \cup V)} + \sum_{u \in U} 2^u$$

Finally, observe that x is not adjacent to anything in V, because for any $v \in V$ the $v - th$ bit in x is zero (since $U \cap V = \emptyset$). □

In view of this lemma, we will consider the following collection of sentences in Ψ. We define for any $k, l \in \mathbb{N}$ the sentence $\phi_{k,l}$ by

$$\begin{aligned} &\phi_{k,l} : \forall x_1, \dots x_k, y_1, \dots y_l \quad \exists z \text{ such that} \\ &e(x_1, z) \wedge \dots e(x_k, z) \wedge \neg e(y_1, z) \dots \wedge \neg e(y_l, z) \end{aligned}$$

where the x_i are distinct from the y_j. We will denote the collection of these sentences by $\Phi = \{\phi_{k,l} | k, l \in \mathbb{N}\}$. This collection satisfies an interesting property.

Lemma 3.5

For any $n \in \mathbb{N}$ and $k, l \in \mathbb{N}$ with $k + l < n$, we have $\mathbb{P}_n(\neg\psi_{k,l}) \leq \binom{n}{k}\binom{n-k}{l}(1 - 2^{-k-l})^{n-k-l}$. In particular $\lim_{n\to\infty}(\neg\psi_{k,l}) = 0$.

Proof. For any fixed collection of $x's$ and $y's$ and any vertex z, the probability that z doesn't satisfy the formula is $(1 - 2^{-k-l})$. There are $n - k - l$ choices for z, so the probability that neither of those satisfies is $(1 - 2^{-k-l})^{n-k-l}$. Now we simply take a union bound over all collections of $x's$ and $y's$. The limit is zero because since k, l are fixed, $\binom{n}{k}, \binom{n-k}{l}$ are polynomials while $(1 - 2^{-k-l})^{n-k-l}$ is exponential. □

A collection of sentences in first-order logic, such as Φ, is called a *theory.* In view of Lemma 3.4, for any $k, l \in \mathbb{N}$, $\phi_{k,l}$ is satisfied on the Rado graph. A structure which satisfied all sentences of a theory is called a *model.* Our hope is that theory Φ is 'powerful enough' so that each sentence in Ψ or its negation can be derived using Φ. This would allows to use Lemma 3.5 to obtain a bound on the limit for sentences in Ψ.

The informal statement that Φ is 'powerful enough' can be more formally defined as Φ being *complete*, which means that any $\psi \in \Psi$ or its negation, can be derived in Φ. Crucially, one can show that Φ is complete, but the proof is beyond the scope of this text. One can also show that Φ is consistent, meaning that it does not imply any contradictions. Both of these follow from the fact that the Rado graph is a *unique* model of Φ.

We are almost ready to prove Theorem 3.3. We need one more theorem from logic.

Theorem 3.6 (Compactness theorem).
For every theory T and every formula ϕ, ϕ is a logical consequence of T if and only if there exists a finite subset T' of T such that ϕ is a logical consequence of T'.

The theorem is usually formulated slightly differently, but this is an equivalent formulation and it is the one we will use.

Proof Of Theorem 3.3. Let $\psi \in \Psi$ be arbitrary. Since Φ is complete, we can derive ψ or its negation using sentences in Φ. Without loss of generality, assume it is ψ. Using the compactness theorem, we only need finitely many sentences in Φ to do so. Let us denote these by $\phi_1, \phi_2 \dots \phi_s$. We have

$$\begin{aligned}
&((\phi_1 \wedge \phi_2 \wedge \dots \phi_s) \to (\psi)) \\
\iff &((\neg\psi) \to (\neg\phi_1 \vee \neg\phi_2 \vee \dots \neg\phi_s)) \\
\implies &\mathbb{P}_n(\neg\psi) \leq \mathbb{P}_n(\neg\phi_1) + \dots + \mathbb{P}_n(\neg\phi_s) \quad \text{(union bound)}
\end{aligned}$$

and thus $\lim_{n\to\infty} \mathbb{P}_n(\neg\psi) = 0$ by Lemma 3.5, and hence $\lim_{n\to\infty} \mathbb{P}_n(\psi) = 1$. This proves the zero-one law of first-order logic. □

3.3.2 Zero-one laws of graph neural networks

MPGNNs can be used to assign binary labels to graphs, and in this case, a connection to sentences in first order logic of graphs emerges. Consider an MPGNN M and let $\mathbb{G}$ be a collection of graphs $G = (V, E, \mathbf{F})$. Using M as a binary classifier, we can view M as a boolean map $M : \mathbb{G} \to \mathbb{B}$. Notice that M behaves similarly to a sentence ψ in first order logic of graphs, in the sense that ψ is also a boolean map over graphs - a graph G is mapped to a label in $\mathbb{B}$ based on the truth value of ψ on G. With an understanding of the zero-one laws of logic, it becomes natural to explore whether MPGNNs adhere to a similar principle. The intriguing idea of exploring zero-one laws of graph neural networks was first presented in [17].

The zero-one law would have important implications about the *class of functions* which can be expressed using using MPGNNs. Consider the following definition

Definition 3.5 ([17]).
Let f be a boolean function on graphs, let M be an MPGNN which assigns binary labels, and let $\delta > 0$. Then, we say M uniformly δ-approximates f if

$$\forall n \in \mathbb{N} : \mathbb{P}(f(G) = M(G) \Big| |G| = n) \geq 1 - \delta$$

if $G \sim \mathbb{G}(n, 1/2)$.

Alternatively, one could view the above definition as requiring that for every n, the proportion of n-node graphs on which $f(G) = M(G)$ is atleast $1 - \delta$. This is because under $G(n, 1/2)$ all graphs on n nodes are equally likely.

If M satisfies a zero-one law, then as $n \to \infty$, M assigns the same label to $G \sim \mathbb{G}(n, 1/2)$, with probability tending to 1. If the function f does not satisfy a zero-one law, meaning both labels are assigned with at least probability $\epsilon > 0$, then the labels assigned by M and f sometimes disagree, and thus there exists a δ for which M does not uniformly δ-approximate f.

Let us illustrate this by an example. Consider the boolean function f

$$f(G) = \begin{cases} 1 & \text{if node 1 has an even number of neighbors} \\ 0 & \text{else} \end{cases}$$

Then clearly for any $n \geq 2$, we have $\mathbb{P}(f(G) = 1 \Big| |G| = n) = 1/2$ if $G \sim \mathbb{G}(n, 1/2)$. Without loss of generality, assume that the labels assigned by M tend to 0, meaning as $n \to \infty$, $\mathbb{P}(M(G) = 0) \to 1$. Now consider the probability that M and f agree as n gets large:

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}(f(G) = M(G) \Big| |G| = n) \\
&= \lim_{n\to\infty} \mathbb{P}(f(G) = 0 \Big| |G| = n)\mathbb{P}(M(G) = 0) + \lim_{n\to\infty} \mathbb{P}(f(G) = 1 \Big| |G| = n)\mathbb{P}(M(G) = 1) \\
&= 1/2 \times 0 + 1/2 \times 1 = 1/2
\end{aligned}$$

thus f and M disagree on half of the graphs, in the limit. This means M does not uniformly δ-approximate f for any $\delta < 1/2$.

Thus, if we could show that MPGNNs indeed satisfy a zero-one law, we could conclude that any boolean function f which *does not* satisfy a zero-one law, cannot be uniformly δ-approximated for some δ by MPGNNs, and this would be a significant expressiveness constraint.

3.3.3 The zero-one law of GCN and MeanGNN

In this section we will give a brief overview of the results of [17]. The paper [17] considers three different models - GCN(2.4), MeanGNN (2.5) and SumGNN (2.6), and proves that under certain conditions, these satisfy a zero-one law. We will only provide an explanation for the first two models. Let us begin by carefully explaining the setup.

Similarly as in the logic case, we will sample graphs G_n of size n from the Erdős-Rényi model $\mathbb{G}(n, p)$, but in addition, we will sample a vector of node features for every node. The feature vectors are sampled *independently* from some distribution $\mathbb{D}$ on $\mathbb{R}^d$, and are stored in an $n \times d$ matrix $\mathbf{F}_n$. Once we have sampled a graph and the node features, we will use a model (either a GCN or a MeanGNN) to assign a label $l \in \mathbb{B} = \{0, 1\}$ to this graph. This is done by attaching a mean pooling layer and a classifier C after the final layer of our MPGNN. The mean pooling layer computes the mean of the final node features produced in the message passing phase, and this mean is then passed into the classifier $C : \mathbb{R}^{d_L} \to \mathbb{B}$, which outputs a label $l \in \mathbb{B} = \{0, 1\}$, where d_L is the output dimension of the final layer of the MPGNN.

This means we can view a model M, as a function which maps a pair $(G_n, \mathbf{F}_n)$ to a label $l \in \mathbb{B}$. We are interested in the probability that our model assigns a particular label in the limit as $n \to \infty$, meaning we want to compute:

$$\lim_{n\to\infty} \mathbb{P}(M((G_n, \mathbf{F}_n)) = 0) = ?$$

and we will conclude that the zero-one law is satisfied, provided that the above equals 0, or 1. The main result of the paper [17] is given by the following theorem:

Theorem 3.7 (Zero-one law of GCN/MeanGNN[17]).
Let M be a GCN/MeanGNN with an arbitrary number of layers, equipped with a mean pooling layer and a classifier $C : \mathbb{R}^{d_L} \to \mathbb{B}$. Let (G_n) be a sequence of graphs sampled from $\mathbb{G}(n, p)$, and let $(\mathbf{F}_n)$ be a sequence o $n \times d$ matrices of node features, sampled independently from a sub-Gaussian distribution D. Then, under the assumption that the classifier C is non-splitting , we have

$$\lim_{n\to\infty} \mathbb{P}(M((G_n, \mathbf{F}_n)) = 0) = 0 \textit{ or } 1$$

In this context, a non-splitting classifier is defined as follows

Definition 3.6
Consider a distribution $\mathbb{D}$ on $\mathbb{R}^d$ with mean $\boldsymbol{\mu}$. Let M be a GCN used for binary graph classification. Define the sequence $\boldsymbol{\mu}^0, \ldots \boldsymbol{\mu}^T$ of vectors inductively by $\boldsymbol{\mu}^0 := \boldsymbol{\mu}$

and $\boldsymbol{\mu}^t := \sigma(\psi^t(\boldsymbol{\mu}^{t-1}) + \mathbf{b}^{t-1})$. *We say the classifier* $C : \mathbb{R}^{d_L} \to \mathbb{B}$ *is non-splitting for* M*, if* $\boldsymbol{\mu}^L$ *does not lie on the decision boundary of* C*.*

Similarly, we could define a non-splitting classifier for MeanGNN (2.5), in the obvious way.

The essence of Theorem 3.7 is that in fact, the output of the mean pooling layer actually converges to $\boldsymbol{\mu}^L$. This is a technical result that uses the fact that D is sub-Gaussian, among other things. Once this is shown, the idea of a non-splitting classifier comes into play. Observe that for any function $C : \mathbb{R}^{d_L} \to \mathbb{B}$, if C is continuous at a point x, then it is locally constant at x, meaning we can find an open set S containing x, such that $\forall y \in S$, $C(y) = C(x)$. This is immediate from the fact that C can only take values in $\mathbb{B}$, and the definition of continuity. Thus, the *decision boundary* of C, is precisely the collection of points where such an open set cannot be formed, which is equivalent to the set of discontinuities.

Now, by Theorem 2.9, if we could show that the output of our model (after mean pooling) converges to $\boldsymbol{\mu}^L$ in distribution (2.14), then the label, obtained by applying the classifier, also converges to $C(\boldsymbol{\mu}^L)$ in distribution. In Theorem 2.9, the assumption is that $\mathbb{P}(X \in D_C) = 0$. In our case however, this assumption becomes trivial - the variable X is the constant $\boldsymbol{\mu}^L$, which either is (with probability 1) on the decision boundary, or is not (with probability 0). So the requirement that $\boldsymbol{\mu}^L$ is not on the decision boundary is precisely the assumption $\mathbb{P}(X \in D_C) = 0$ in Theorem 2.9. This is why we require the value $\boldsymbol{\mu}^L$ to not lie on the decision boundary of C, i.e in the set of discontinuities. With this assumption we can simply conclude, by Theorem 2.9, that the label converges to $C(\mu^L)$ (which is a constant, either 0 or 1), in distribution. This would yield the zero-one law, because it would mean that

$$\lim_{n\to\infty} \mathbb{P}(M((G_n, \mathbf{F}_n)) = C(\boldsymbol{\mu}^L)) = 1$$
$$\lim_{n\to\infty} \mathbb{P}(M((G_n, \mathbf{F}_n)) \neq C(\boldsymbol{\mu}^L)) = 0$$

which is exactly the desired zero-one law. This is exactly why we require the classifier to be *non-splitting.*

However, the real difficulty is showing that the output of the model converges to $\boldsymbol{\mu}^L$. Instead of showing the proof from the paper, let us give some motivation why one should expect this to be true on an intuitive level.

According to the Laws of Large Numbers (2.8), the sample mean of independent and identically distributed (i.i.d) random variables converges - almost surely (2.13) - towards the mean of the corresponding distribution, provided that the mean exists. Note that the aggregation scheme of GCN (2.4) and MeanGNN (2.5) does not precisely mirror the sample mean, but it bears a noteworthy resemblance. To simplify things and illuminate the concept, consider a message passing network employing the following aggregation scheme, similar to that of a MeanGNN:

$$\mathbf{h}_u^t = \frac{1}{|N(u)|} \sum_{v \in N(u)} \mathbf{h}_v^{t-1}$$

Assuming that the $\mathbf{h}_v^{t-1}$ are independent random variables, we could infer that $\mathbf{h}_u^t$ converges - either almost surely or in distribution - towards the mean of the underlying distribution, denoted by $\boldsymbol{\mu}$. In the first layer, $\mathbf{h}_v^{t-1}$ are just the initial node features, which are independent by assumption. This means that for every node, the node features obtained after the first layer converge. Now, if we applied a mean pooling layer after the first layer, one would expect its output to also converge to $\boldsymbol{\mu}$. This means that the label of the graph should also converge to $C(\boldsymbol{\mu})$ (under the non-splitting assumption), which would yield the result (for the first layer). In the case of more layers, one can imagine that the result would follow by an inductive argument, because the input to the second layer already converges. This is indeed the case, and induction is used in the actual proof.

However, to actually prove the zero-one law one must obtain explicit bounds on $\mathbb{P}(M(G_n, \mathbf{F}_n))$ in order to compute the limit. This is non-trivial and further assumptions are needed, such as the features being sub-Gaussian. Thus, the above is a mere intuitive explanation of the phenomenon, and we will see a more rigorous proof in the next section.

4
Exploring zero-one laws of attention networks

In the previous section, we observed that GCN and MeanGNNs satisfied the zero-one law for graphs sampled from the Erdős-Rényi model. A natural next step is to try extend these results to other models. In recent years, models which aggregate using a weighted mean of the neighborhood, such as graph attention networks [22], have become increasingly popular, and therefore it is natural to ask whether the expressivity limitation described by the zero-one law also applies to this class of models.

Attempting to extend the zero-one law to models with an attention mechanism was the main goal of this thesis. This problem turns out to be hard. The introduction of the attention mechanism makes the situation significantly more complicated from a theoretical point of view, and more sophisticated techniques are necessary.

Recall that in the previous section, we discussed the connection between the zero-one law and the laws of large numbers of probability theory. Even though the proof did not use laws of large numbers, this connection allowed us to gain intuition about the problem, and 'guess' what the limiting behaviour should be. However, in the case of attention networks, this connection is not as straightforward, and therefore it was challenging to predict the behaviour of such models. Because of this, we began by considering a simpler version of attention networks, mainly to gain intuition into the convergence behaviour of weighted sums.

Thus, in the next chapter we will consider a slightly modified problem, one which actually is not that relevant from a machine learning perspective, but is of a more mathematical nature, and will hopefully provide the reader with some insights into the convergence of weighted sums. We will deal with actual models, such as graph attention networks and transformers, in later chapters, and that is where the situation will become very technical.

4.1 First steps

Attention networks, are models which have an aggregation scheme given by

$$\mathbf{h}_u^{l+1} = \sum_{v \in N^+(i)}^{n} \alpha_{uv} \psi^l(\mathbf{h}_v^l)$$

where α_{ij} is the attention weight, and ψ^l is a linear transformation. The choice of the attention mechanism varies from model to model, and examples include the Transformer (2.8), GAT (2.9) and GATv2 (2.10). Similarly as before, the question that we are considering is whether these models converge as the graph size increases, provided that the node features are sampled randomly, and whether the zero-one law is satisfied.

The first thing to notice is that the aggregation defined above is a generalization of the models that we considered previously. For example, upon setting α_{uv} equal to $1/\sqrt{|N(u)||N(u)|}$, one obtains the GCN. Thus we can immediately conclude that in some cases, the zero-one law is satisfied.

On the other hand, it is quite clear that there exist situations where convergence definitely does not occur. As was discussed previously, we know that the essence of the zero-one law is the fact that $\mathbf{h}_u^1$ converges. But this can easily be violated — for example, if the attention function was such that it discards everything but a node itself:

$$\alpha_{uv} = \begin{cases} 1 & u = v \\ 0 & \text{else} \end{cases}$$

then the output is just $\mathbf{h}_u^1 = \psi^1(\mathbf{x_u})$, which is just a random variable which does not converge anywhere (unless it is a constant). This means that the convergence behaviour heavily depends on what kind of attention mechanism we choose, and it is not immediately clear where the 'boundary' is.

There is a lot of freedom in how to choose the attention mechanism. For example, we can assume that the attention weights are random variables which are independent from the node features, as is the case in the case of GCN or MeanGNN. However, this is not what is usually meant by attention networks. Usually, the whole point of attention is to compute it based on the node features. For example, GATs compute attention as

$$\alpha_{ij} = \mathit{softmax}_j(\text{LeakyReLU}(\langle \mathbf{a}^l, \psi^l(\mathbf{x}_i^l) || \psi^l(\mathbf{x}_j^l) \rangle))$$

where $||$ denotes concatenation, $\mathbf{a}$ is a vector and ψ^l a linear map (matrix), and both are learnable parameters. Models where the attention is a function of the features are much more difficult to analyze, but also much more interesting.

However, even then, there is something that most attention mechanisms have in common. In GCN/MEANGNN, the attention is resembling $1/n$, where n is the size of the neighborhood. In attention models which use a softmax, one would expect the attention weight to be close to $1/n$ on average, and it is quite unlikely that one would see large deviations in the attention from this value. This is not a rigorous statement, but it should be intuitively clear. This suggests that perhaps, attention models which use a softmax could also satisfy the zero-one law.

This observation motivates the following question - suppose the attention coefficients are such that they are 'close' to $1/n$, but behave similarly to the softmax, in the sense that they must sum to one. Would the zero one-law still be satisfied or not? If yes, how wildly can the weights oscillate around their mean of $1/n$, while still preserving convergence?

4.2 Dirichlet-GNN: Attention networks with Dirichlet distribution

One way to study the problem described above, is to place a distribution on the attention weights and study the convergence with respect to that distribution. In view of what was said previously, we would like the distribution to produce weights which are close to $1/n$, but not necessarily equal, and they sum to 1. For this, we can use the Dirichlet distribution.

The Dirichlet distribution with k parameters, denoted $D(\alpha_1, \dots \alpha_k)$ produces k values $x_1, x_2, \dots, x_k$, such that $x_i \in [0,1]$, and $\sum_{i=1}^{k} x_i = 1$. Letting $\alpha_0 = \sum_{i=1}^{n} \alpha_i$. For any i, we have

$$\mathbb{E}[x_i] = \frac{\alpha_i}{\alpha_0}$$
$$Var[X_i] = \frac{\frac{\alpha_i}{\alpha_0}(1 - \frac{\alpha_i}{\alpha_0})}{\alpha_0 + 1}$$

In our case, we will mostly be interested in the case when the expectation is $1/n$, which is achieved when we set $\alpha_1 = \alpha_2 \dots \alpha_n = q$. In this case, the variance becomes

$$Var[X_i] = \frac{n-1}{n^2(nq+1)} = O(1/n^2)$$

This setup allows us to have an attention mechanism with weights summing to 1 and being relatively close to $1/n$. Thus, in this section we will study the following model

Definition 4.1

A Dirichlet-GNN is a MPGNN which aggregates as $\mathbf{x}_u^l = \sigma(\mathbf{y}_u^l)$, where σ is a non-linearity of choice (such as ReLU) and

$$\mathbf{y}_u^{l+1} = \sum_{v \in N^+(u)} \alpha_{u,v}^l \psi^l(\mathbf{x}_v^l) + \mathbf{b}^l$$

where ψ^l is a linear map and $(\alpha_{u,v_1}^l, \alpha_{u,v_2}^l \dots \alpha_{u,v_{|N^+(u)|}}^l) \sim D(q^l, \dots q^l)$ for some $q^l > 0$.

Note that this particular model is probably quite useless from a machine learning perspective, in the sense that in the above definition, the attention weights are sampled from the Dirichlet distribution, and are not computed based on the input graph or features. However, in our discussion above we illustrated that many attention models produce attention weights whose distribution is 'similar' to the Dirichlet distribution. Therefore, understanding the convergence behaviour of Dirichlet-GNN would provide insights into the asymptotic behaviour of actual attention models. For this reason, the goal of the following sections will be to prove the following theorem:

Theorem 4.1 (Zero-one law of Dirichlet-GNN).
Let M be a Dirichlet-GNN equipped with a final mean-pooling layer, a non-splitting classifier, and a non-linearity σ between the layers, which is c-Lipschitz continuous. Let (G_n) be a sequence of graphs sampled from $\mathbb{G}(n, p)$ and $(\mathbf{F}_n)$ a sequence of matrices of size $n \times d$, containing node features sampled independently from a distribution $\mathbb{D}$ with mean $\boldsymbol{\mu}$. Assume that $\mathbb{D} - \boldsymbol{\mu}$ is sub-Gaussian. Then

$$\lim_{n\to\infty} \mathbb{P}(M(G_n, \mathbf{F}_n) = 0) = 0 \text{ or } 1$$

4.3 Dirichlet random walks

Before we prove the zero-one law, we must obtain a better theoretical understanding of Dirichlet random walks, defined as

$$S_n = \alpha_1^n X_1 + \alpha_2^n X_2 + \dots \alpha_n^n X_n$$

where the weights α_i^n come from $D(q_1, q_2 \dots q_n)$.

We know, by the laws of large numbers, that the sample mean of integrable, independent and identically distributed random variables converges to its mean. But what happens if it is a weighted mean, where the weights sum to 1. Does convergence occur? If yes, is it still convergence to the mean? This question is interesting purely from a mathematical point of view and the answer will determine whether a zero-one law occurs or not.

Surprisingly, there is very little literature about this problem, but some related works are [42, 43]. Let us now give a first result. The following theorem tells us that S_n indeed converges to the mean, as n goes to infinity.

Theorem 4.2

Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2 \ldots \alpha_n) \sim D(q, q, \ldots, q)$. Let (X_i) be i.i.d with mean μ and variance σ^2. Assume X_i, α_j are independent for all i, j. Then $S_n \to \mu$ in distribution.

Proof. First observe that $\mathbb{E}[S_n] = n \times \frac{\mu}{n} = \mu$, using properties of the Dirichlet distribution. Now, we would like to apply Chebyshev's inequality 2.6. For this we need to find the variance. We have:

$$\begin{aligned} Var[S_n] &= \sum_{n=1}^{n} Var[\alpha_i X_i] + 2\sum_{i \neq j} Cov(\alpha_i X_i, \alpha_j X_j) \\ &= n \times Var[\alpha_1 X_1] + 2 \times \binom{n}{2} Cov(\alpha_1 X_1, \alpha_2 X_2) \end{aligned}$$

Let's proceed step by step. For the first term, we have:

$$\begin{aligned} Var[\alpha_1 X_1] &= \mathbb{E}[\alpha_1^2 X_1^2] - \frac{\mu^2}{n^2} \\ &= \mathbb{E}[\alpha_1^2]\mathbb{E}[X_1^2] - \frac{\mu^2}{n^2} \end{aligned}$$

Using properties of the Dirichlet distribution, we have:

$$\begin{aligned} \mathbb{E}[\alpha_1^2] &= Var[\alpha_1] + \frac{1}{n^2} \\ &= \frac{\frac{q}{nq}(1 - \frac{q}{nq})}{nq + 1} + \frac{1}{n^2} \\ &= \frac{\frac{1}{n}(1 - \frac{1}{n})}{nq + 1} + \frac{1}{n^2} \end{aligned}$$

And $\mathbb{E}[X_1^2] = \mu^2 + \sigma^2$. Combining these, we obtain:

$$\begin{aligned} Var[\alpha_1 X_1] &= \left(\frac{\frac{1}{n}(1 - \frac{1}{n})}{nq + 1} + \frac{1}{n^2} \right) \times (\mu^2 + \sigma^2) - \frac{\mu^2}{n^2} \\ &= \frac{\frac{1}{n}(1 - \frac{1}{n})}{nq + 1} \times (\mu^2 + \sigma^2) + \frac{\sigma^2}{n^2} \end{aligned}$$

Now, we need to compute the covariance term:

$$\begin{aligned} Cov(\alpha_1 X_1, \alpha_2 X_2) &= \mathbb{E}\left[(\alpha_1 X_1 - \frac{\mu}{n})(\alpha_2 X_2 - \frac{\mu}{n}) \right] \\ &= \mathbb{E}[\alpha_1 X_1 \alpha_2 X_2] - \frac{\mu^2}{n^2} \\ &= \mu^2 \mathbb{E}[\alpha_1 \alpha_2] - \frac{\mu^2}{n^2} \end{aligned}$$

Now we use the properties of the Dirichlet distribution to get:

$$\begin{aligned}\mathbb{E}[\alpha_1\alpha_2] &= Cov(\alpha_1, \alpha_2) + \frac{1}{n^2}\\ &= -\frac{\frac{q}{nq} \times \frac{q}{nq}}{nq+1} + \frac{1}{n^2}\\ &= \frac{1}{n^2}(1 - \frac{1}{nq+1})\end{aligned}$$

This gives:

$$\begin{aligned}Cov(X_1\alpha_1, X_2\alpha_2) &= \frac{\mu^2}{n^2}(1 - \frac{1}{nq+1}) - \frac{\mu^2}{n^2}\\ &= -\frac{\mu^2}{n^2} \times \frac{1}{nq+1}\end{aligned}$$

Now finally, we obtain:

$$\begin{aligned}Var[S_n] &= n \times \frac{\frac{1}{n}(1-\frac{1}{n})}{nq+1} \times (\mu^2+\sigma^2) + \frac{\sigma^2}{n^2} + 2\frac{n(n-1)}{2}\frac{\mu^2}{n^2} \times \frac{-1}{nq+1}\\ &= \sigma^2\left(\frac{1-\frac{1}{n}}{nq+1} + \frac{1}{n^2}\right) + \mu^2\left(\frac{1-\frac{1}{n}}{nq+1} - \frac{1-\frac{1}{n}}{nq+1}\right)\\ &= \sigma^2\left(\frac{1}{n}(\frac{n-1}{nq+1}+1)\right)\\ &= \sigma^2\frac{q+1}{nq+1}\end{aligned}$$

Now we can apply Chebyshev's inequality. For a fixed $n \in \mathbb{N}$, we have:

$$\begin{aligned}\mathbb{P}(|S_n - \mu| > \epsilon) &\leq \frac{Var[S_n]}{\epsilon^2}\\ &= \frac{\sigma^2(q+1)}{\epsilon^2(nq+1)} \to 0 \text{ as } n \to \infty\end{aligned}$$

Thus S_n converges to μ in distribution. □

Although the computation is long, it is not difficult and the final formula for the variance can be easily verified experimentally. This is done in Figure 4.1 - the blue curve represents a numerically computed variance of S_n, for different values of n. The features X_i were sampled from a normal distribution with mean equal to 1.2 and standard deviation equal to 3.4 (these were chosen to be some arbitrary values). The variance was computed from 1000 samples, for each n. The orange curve is the plot of $(3.4)^2\frac{q+1}{nq+1}$.

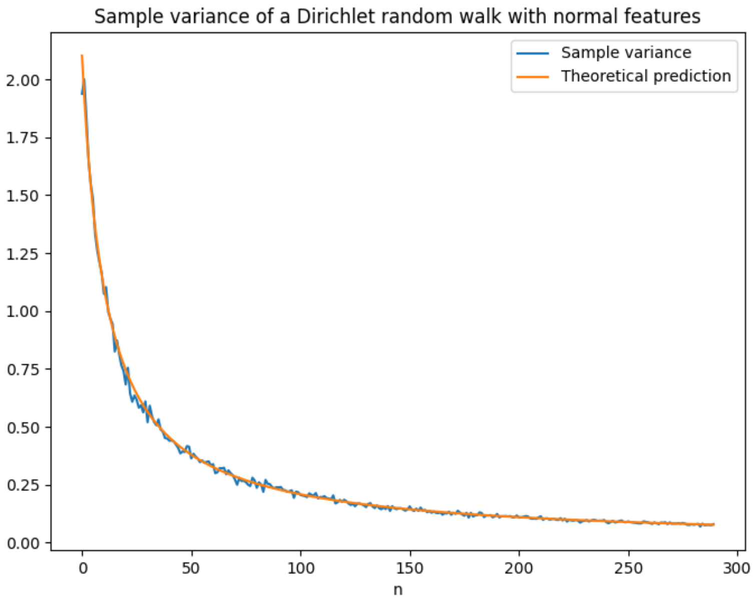


Figure 4.1: A comparison of the computed formula for $Var[S_n]$, and the sample variance of S_n. The sample variance closely tracks the computed curve.

Recall that we are exploring these random walks with the zero-one laws in mind. We know that in order to prove it, we would need the convergence to happen for all nodes simultaneously. This means we would take a union bound over all n nodes in the last step of the proof. In order to obtain a bound which goes to zero even after taking a union bound, we can place some restrictions on the features X_i. Similarly as in the case of GCN and MEANGNN, it helps if we assume that the features are sub-Gaussian (2.11). It turns out that once we assume the vectors are sub-Gaussian, we can obtain interesting concentration bounds. Firstly, we have the following useful lemma.

Lemma 4.3

Let $X_1, \ldots X_n$ be i.i.d subG(σ^2). Let $\boldsymbol{\alpha} = (\alpha_1 \ldots \alpha_n)$ be a random vector, with α_i independent from the X_j for all i, j. Then

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i| > t) \leq 2\mathbb{E}_\alpha \left[\exp\left(\frac{-t^2}{\sigma^2 ||\boldsymbol{\alpha}||_2^2} \right) \right]$$

Proof. We can use the continuous law of total probability. Let $f(\boldsymbol{\alpha})$ denote the pdf of $\boldsymbol{\alpha}$:

$$\begin{aligned}
\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i|> t) &= \int \mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i|> t|\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n))f(\boldsymbol{\alpha})d\boldsymbol{\alpha} \\
&\leq \int 2\exp\left(\frac{-t^2}{\sigma^2||\boldsymbol{\alpha}||_2^2}\right) f(\boldsymbol{\alpha})d\boldsymbol{\alpha} \\
&= 2\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\right]
\end{aligned}$$

where we used Lemma 2.5. □

In our case, the random variables have mean μ, which is not necessarily 0 and the random vector comes from the Dirichlet distribution. In order to apply Lemma 4.3 and obtain a useful bound, we can proceed as follows. First, in view of Definition 2.11, we require $X_i - \mu$ to be sub-Gaussian. Now, we can define $Y_i = X_i - \mu$, and since Y_i are sub-Gaussian random variables, we can apply Lemma 4.3. Finally, recall that since the coefficient α come from the Dirichlet distribution, they must sum to 1. Thus, we have

$$\sum_{i=1}^{n} \alpha_i Y_i = \sum_{i=1}^{n} \alpha_i X_i - \mu$$

and thus

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i Y_i|> t) = \mathbb{P}(|S_n - \mu|> t)$$

Now we can apply Lemma 4.3 to $\mathbb{P}(|\sum_{i=1}^{n} \alpha_i Y_i|> t)$, we get

$$\mathbb{P}(|S_n - \mu|> t) \leq 2\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\right]$$

Now, our goal is to show that this expectation decays sufficiently quickly with n. Note that the dependency on n is hidden in the term $\|\boldsymbol{\alpha}\|_2^2$, which is an n dimensional vector, sampled from the Dirichlet distribution. In order to understand the asymptotic behaviour of the above bound, we need some auxilliary results.

Lemma 4.4

Let $\mathbf{x} = (x_1 \dots x_n) \sim D(\alpha_1, \alpha_2 \dots \alpha_n)$. *Then*

$$\mathbb{E}[\|\mathbf{x}\|_2^2] = \frac{\sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \alpha_i^2}{(1 + \sum_{i=1}^{n} \alpha_i)(\sum_{i=1}^{n} \alpha_i)}$$

Proof. Let $Q_n = \{\mathbf{x} \in \mathbb{R}^n | \|x\|_1 = 1\}$. The pdf of the Dirichlet distribution is given by $f(x_1, x_2 \ldots x_n, \alpha_1 \ldots \alpha_n) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^n x_i^{\alpha_i - 1}$, where $(x_1, \ldots x_n) \in Q_n$, $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_n)$ and $B(\boldsymbol{\alpha}) = \frac{\prod \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}$, where Γ denotes the gamma function. Now

$$\begin{aligned}\mathbb{E}[\|x\|_2^2] &= \frac{1}{B(\boldsymbol{\alpha})} \int_{Q_n} (x_1^2 + \ldots x_n^2) \prod_{i=1}^n x_i^{\alpha_i - 1} d\mathbf{x} \\ &= \sum_{j=1}^n \int_{Q_n} \prod_{i=1}^n x^{\alpha_i - 1 + 2\delta_{ij}} d\mathbf{x}\end{aligned}$$

where δ_{ij} denotes the Kronecker delta. But using the pdf of the Dirichlet distribution, this is just:

$$\begin{aligned}&= \sum_{j=1}^n \frac{1}{B(\boldsymbol{\alpha})} \int_{Q_n} \prod_{i=1}^n x^{\alpha_i - 1 + 2\delta_{ij}} d\mathbf{x} \\ &= \sum_{j=1}^n \frac{B(\alpha^i)}{B(\boldsymbol{\alpha})}\end{aligned}$$

where $B(\alpha^i) = \frac{\prod \Gamma(\alpha_i + 2\delta_{ij})}{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}$. Now using the property of the gamma function that $\Gamma(z+1) = z\Gamma(z)$, we get that

$$\begin{aligned}\frac{B(\alpha^i)}{B(\boldsymbol{\alpha})} &= \frac{\frac{\prod \Gamma(\alpha_i + 2\delta_{ij})}{\Gamma\left(2 + \sum_{i=1}^k \alpha_i\right)}}{\frac{\prod \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}} \\ &= \frac{(\alpha_j + 1)\alpha_j}{(1 + \sum_{i=1}^n \alpha_i) \sum_{i=1}^n \alpha_i}\end{aligned}$$

and finally, we obtain the result simply by summing over j. □

In particular, in the case where $\alpha_i = q$ for all i, we get that $\mathbb{E}[\|\mathbf{x}\|_2^2] = \frac{1+q}{1+nq}$, which can easily be verified experimentally, as is done in Figure 4.2.

Note that in the case of $\alpha_i = q$, $\mathbb{E}[\|\mathbf{x}\|_2^2] = \frac{1+q}{1+nq}$, tends to 0 as n tends to infinity, which gives us hope that maybe, $\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2 \|\boldsymbol{\alpha}\|_2^2}\right)\right]$ also tends to 0 as n tends to infinity, because observe that $\lim_{x \to 0} \exp \frac{-t^2}{\sigma^2 x} = 0$. However, the argument is not this straightforward - what we really need is $\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2 \|\mathbf{x}\|_2^2}\right)\right]$. If we could pull the expectation inside the exponential, we would be done. Unfortunately, we cannot do this, not even using Jensen's inequality because the function is not convex/concave.

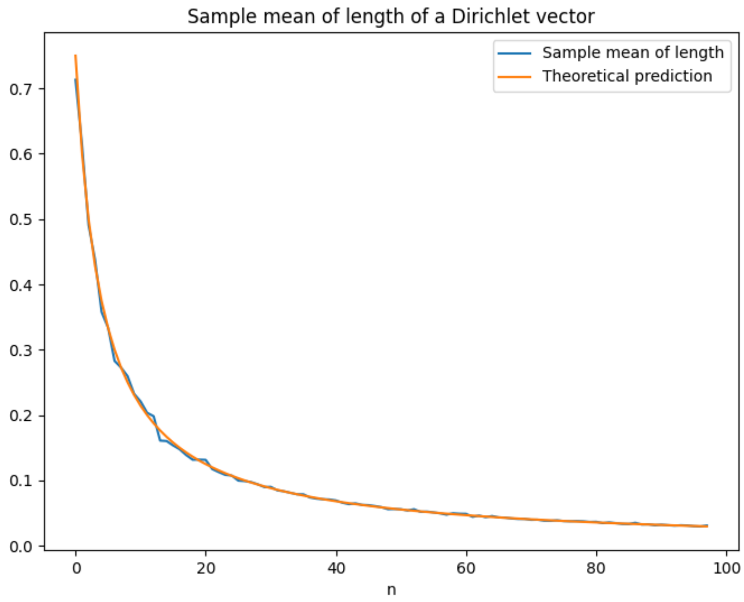


Figure 4.2: A comparison of the computed formula for $\mathbb{E}[\|\mathbf{x}\|_2^2]$, and the sample mean of $\|\mathbf{x}\|_2^2$. The sample mean closely tracks the computed curve.

The idea is to prove one more concentration bound, and deduce that $\|\mathbf{x}\|_2^2$ is close to its expectation with high probability. To find whether this is the case, we will compute the variance of $\|\mathbf{x}\|_2^2$ and utilize Chebyshev's inequality.

Lemma 4.5

Let $\mathbf{x} = (x_1, \dots x_n) \sim D(q, q, \dots, q)$. *Then*

$$Var[\|\mathbf{x}\|_2^2] = \frac{2(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)}$$

Proof. Let $\mathbf{q} = (q, \dots, q) \in \mathbb{R}^n$. We have $Var[\|\mathbf{x}\|_2^2] = \mathbb{E}[\|\mathbf{x}\|_2^4] - (\mathbb{E}[\|\mathbf{x}\|_2^2])^2$. We already know the second term, so we only need to find the first one. We have:

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}\|_2^4] &= \frac{1}{B(\mathbf{q})}\int_{Q_n}(x_1^2 + \dots x_n^2)^2 \prod_{i=1}^n x_i^{q-1} d\mathbf{x} \\ &= \sum_{i=1}^n \frac{1}{B(\mathbf{q})}\int_{Q_n} x_i^4 \prod_{j=1}^n x_j^{q-1} d\mathbf{x} + 2\sum_{i\neq j}\frac{1}{B(\mathbf{q})}\int_{Q_n} x_i^2 x_j^2 \prod_{k=1}^n x_k^{q-1} d\mathbf{x} \\ &= \sum_{i=1}^n \frac{B^i(q)}{B(\mathbf{q})} + 2\sum_{i\neq j}\frac{B^{ij}(q)}{B(\mathbf{q})}\end{aligned}$$

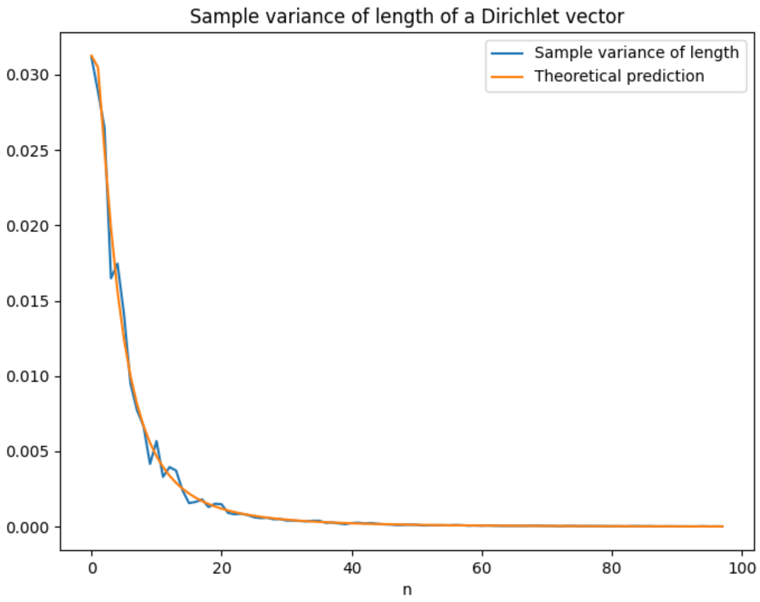


Figure 4.3: A comparison of the computed formula for $Var[\|\mathbf{x}\|_2^2]$, and the sample variance of $\|\mathbf{x}\|_2^2$. The sample variance closely tracks the computed curve.

where

$$B^i(q) = \prod_{j=1}^{n} \Gamma(q + 4\delta_{ij})/\Gamma(4 + nq)$$

$$B^{ij}(q) = \prod_{k=1}^{n} (q + 2\delta_{ik} + 2\delta_{jk})/\Gamma(4 + nq)$$

Using the properties of the gamma function and summing we have:

$$\mathbb{E}[\|\mathbf{x}\|_2^4] = \frac{(n-1)(1+q)^2 q + (3+q)(2+q)(1+q)}{(3+nq)(2+nq)(1+nq)}$$

which means:

$$\begin{aligned} Var[\|\mathbf{x}\|_2^2] &= \frac{(n-1)(1+q)^2 q + (3+q)(2+q)(1+q)}{(3+nq)(2+nq)(1+nq)} - \frac{(1+q)^2}{(1+nq)^2} \\ &= \frac{2(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)} \end{aligned}$$

□

Once again, this can easily be verified experimentally, as is done in Figure 4.3. Observe that the variance decays as $1/n^3$, which means $\|\mathbf{x}\|_2^2$ is very tightly concentrated around the mean. Now we can use Chebyshev's inequality to get the following lemma:

Lemma 4.6

Let $\mathbf{x} \sim D(q, q, \ldots, q)$. *Then*

$$\mathbb{P}\left(\left|\|\mathbf{x}\|_2^2 - \mathbb{E}[\|\mathbf{x}\|_2^2]\right| > \frac{1}{\sqrt{n}}\right) \leq n \times \frac{2(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)}$$

Proof. Apply Chebyshev's inequality 2.6 using the formula for the variance from the previous lemma. □

Now we have the following key result:

Theorem 4.7

Let $X_1 \ldots X_n$ *be independent, sub-gaussian random variables with mean zero and variance proxy* σ^2. *Let* $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_n) \sim D(q, q, \ldots, q)$ *be a random vector independent from the* X_i. *Then for any* $t > 0$:

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i| > t) \leq 2\exp\left(\frac{-t^2}{\sigma^2}\left(\frac{n\sqrt{n}q + \sqrt{n}}{q(n+\sqrt{n}) + \sqrt{n} + 1}\right)\right) + \frac{4n(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)}$$

Proof. By Lemma 4.3, we have:

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i| > t) \leq 2\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\right]$$

Observe that the function $f(\|\boldsymbol{\alpha}\|_2^2) = \exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)$ is increasing, and since $\boldsymbol{\alpha}$ comes from a Dirichlet distribution, f is maximized when $\|\boldsymbol{\alpha}\|_2^2 = 1$. Furthermore, f is always less than or equal to 1. We can now use iterated expectation and apply this observation:

$$\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\right] =$$
$$\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\middle|\left|\|\boldsymbol{\alpha}\|_2 - \mathbb{E}[\|\boldsymbol{\alpha}\|_2^2]\right| > \frac{1}{\sqrt{n}}\right] \times \mathbb{P}\left(\left|\|\boldsymbol{\alpha}\|_2 - \mathbb{E}[\|\boldsymbol{\alpha}\|_2^2]\right| > \frac{1}{\sqrt{n}}\right) +$$
$$\mathbb{E}\left[\exp\left(\frac{-t^2}{\sigma^2\|\boldsymbol{\alpha}\|_2^2}\right)\middle|\left|\|\boldsymbol{\alpha}\|_2 - \mathbb{E}[\|\boldsymbol{\alpha}\|_2^2]\right| \leq \frac{1}{\sqrt{n}}\right]\mathbb{P}\left(\left|\|\boldsymbol{\alpha}\|_2 - \mathbb{E}[\|\boldsymbol{\alpha}\|_2^2]\right| \leq \frac{1}{\sqrt{n}}\right)$$
$$\leq 1 \times \mathbb{P}\left(\left|\|\boldsymbol{\alpha}\|_2 - \mathbb{E}[\|\boldsymbol{\alpha}\|_2^2]\right| > \frac{1}{\sqrt{n}}\right) + \exp\left(\frac{-t^2}{\sigma^2(\mathbb{E}[\|\boldsymbol{\alpha}\|_2^2] + \frac{1}{\sqrt{n}})}\right) \times 1$$

Now, the first term can be bounded using the variance, as was done in the previous lemma. For the second term, recall that $\mathbb{E}[\|\boldsymbol{\alpha}\|_2^2] = \frac{q+1}{nq+1}$. Plugging these in, we obtain:

$$\mathbb{P}(|\sum_{i=1}^{n} \alpha_i X_i|> t) \leq \frac{4n(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)} + 2\exp\left(\frac{-t}{\sigma^2}\left(\frac{n\sqrt{n}q+\sqrt{n}}{q(n+\sqrt{n})+\sqrt{n}+1}\right)\right)$$

as desired. □

Clearly, this bound tends to 0 as n goes to infinity. Furthermore, it goes to zero even if we multiply it by n. This is crucial, because now we can estimate the probability that the sum is small enough at all nodes of our graph at the same time.

Theorem 4.8

Let $X_1 \ldots X_n$ be sub-gaussian random variables with mean μ. Let $\boldsymbol{\alpha^1}, \boldsymbol{\alpha^2} \ldots, \boldsymbol{\alpha^n} \sim D(q, \ldots, q)$ be n attention vectors (one for each node), independent from each other and independent from the X_i. Then for $t > 0$ we have:

$$\mathbb{P}(\exists \text{ a } j \text{ with } |\sum_{i=1}^{n} \alpha_i^j X_i - \mu|> t) \to 0 \text{ as } n \text{ tends to } \infty.$$

Proof. If we let $Y_i = X_i - \mu$, then Y_i are sub-gaussian with mean zero. Now by a union bound and the previous theorem, we have:

$$\begin{aligned}\mathbb{P}(\bigcup_{j=1}^{n}|\sum_{i=1}^{n} \alpha_i^j X_i - \mu|> t) &= \mathbb{P}\left(\bigcup_{j=1}^{n}|\sum_{i=1}^{n} \alpha_i^j Y_i|> t\right) \\ &\leq n\mathbb{P}\left(\sum_{i=1}^{n}|\alpha_i^1 Y_i|> t\right) \\ &\leq 2n \times \exp\left(\frac{-t}{\sigma^2}\left(\frac{n\sqrt{n}q+\sqrt{n}}{q(n+\sqrt{n})+\sqrt{n}+1}\right)\right) \\ &+ n \times \frac{4n(n-1)q(q+1)}{(nq+1)^2(nq+2)(nq+3)}\end{aligned}$$

which clearly tends to 0 as n tends to infinity. □

4.4 A zero-one law for Dirichlet-GNN

So far, all the lemmas and theorems in this section have been presented in a purely mathematical fashion, and are perhaps of independent interest. Now, we would like to use these to prove Theorem 4.1 and obtain a zero-one law for Dirichlet-GNN.

Now that we have the bound from Theorem 4.7, the proof of Theorem 4.1 is just a relatively simple application of this bound. In particular, we can use a similar technique as in the proof of GCN, in the paper by [17]. All we really need to do is show that each node has 'enough' edges. This is done by using a well-known bound (Theorem 2.10) on the Binomial distribution.

Proof of Theorem 4.1. Throughout this proof, we will use notation as in Definition 4.1.

Let us denote the node features by $\mathbf{x}_1 \dots \mathbf{x}_n$. By assumption $\mathbf{x}_i - \boldsymbol{\mu}$ are sub-Gaussian. Further, note that if ψ is a linear map, then $[\psi(\mathbf{x}_i - \boldsymbol{\mu})]_j$, where j denotes the j-th component, is sub-Gaussian, which follows directly from Definition 2.12. We begin by looking at the pre-activations of the first layer.

Since G_n comes from $\mathbb{G}(n,p)$, we know that the $|N^+(v)|$ follow a $1 + Bin(n,p)$ distribution (the $+1$ comes from the self loop), which has an expected value of $1 + np$. By a Chernoff bound 2.10, with $\delta = 1/2$, we have, for any node $u \in V(G)$,

$$\mathbb{P}(|N(u)|\leq \frac{np}{2}) \leq \exp(-\frac{np}{8})$$

And thus

$$\mathbb{P}(|N^+(u)|\leq \frac{np}{2}) \leq \exp(-\frac{np}{8})$$

thus, by taking a union bound we have

$$\mathbb{P}(\exists \text{ node } u : |N^+(u)|\leq \frac{np}{2}) \leq n\exp(-\frac{np}{8})$$

and the complementary event therefore satisfies

$$\mathbb{P}(\forall u \in V : |N^+(u)|\geq \frac{np}{2}) \geq 1 - n\exp(-\frac{np}{8})$$

Now consider a fixed node u, and assume that $|N^+(u)|\geq \frac{1}{2}np$. Then by the bound obtained in Theorem 4.7, we have

$$\mathbb{P}\left([\sum_{v\in N^+(u)} \alpha_{u,v}\psi^1(\mathbf{x}_v) - \psi^1(\boldsymbol{\mu})]_i > t\right) \leq \exp\left(\frac{-t^2}{\sigma^2}\left(\frac{\frac{1}{2}np\sqrt{\frac{1}{2}npq} + \sqrt{\frac{1}{2}np}}{q(\frac{1}{2}np + \sqrt{1/2np}) + \sqrt{\frac{1}{2}np} + 1}\right)\right) \\ + \frac{2\frac{1}{2}np(\frac{1}{2}np - 1)q(q+1)}{(\frac{1}{2}npq + 1)^2(\frac{1}{2}npq + 2)(\frac{1}{2}npq + 3)}$$

Let $B(t)$ denote the bound on the right hand side. Once we pass $\mathbf{y}_u^1$ through the non-linearity, we can use the Lipschitz condition to get:

$$\mathbb{P}(|\mathbf{x}_u^1 - \sigma(\psi^1(\boldsymbol{\mu}))|_i > t) \leq \mathbb{P}(|\mathbf{y}_u^1 - \psi^1(\boldsymbol{\mu}))|_i > \frac{t}{c}) \leq B(\frac{t}{c})$$

where c is the Lipschitz constant of σ.

Now, conditioning on the fact $\forall u : |N^+(u)| \geq \frac{1}{2}np$, and using the law of total probability, we get that

$$\mathbb{P}\left(\exists \text{ node } u : |\mathbf{x}_u^1 - \sigma(\psi^1(\boldsymbol{\mu}))|_i > t\right) \leq 1 \times n \times B(\frac{t}{c}) + n \exp(-\frac{np}{8}) \times 1$$

where we took a union bound in the first term. The right hand side tends to zero as $n \to \infty$ (Theorem 4.8).

Now we want to obtain a bound on every component i. For this we can take a union bound over the dimension d of the node features

$$\mathbb{P}\left(\cup_{i=1}^d \exists \text{ node } u : |\mathbf{x}_u^1 - \sigma(\psi^1(\boldsymbol{\mu}))|_i > t\right) \leq d\left(1 \times n \times B(\frac{t}{c}) + n \exp(-\frac{np}{8}) \times 1\right)$$

and the right hand side tends to 0 as n goes to infinity. This concludes the first layer. Let us now consider the second layer.

We condition on the fact that for every node and every i, we have $|\boldsymbol{x}_u^1 - \sigma(\psi^1(\boldsymbol{\mu}))|_i < \frac{\epsilon}{\|\psi^2\|_\infty}$. We know the probability of this being the case tends to 1 with n, from our analysis of the first layer. If we define $\mathbf{z}_1 = \sigma(\psi^1(\boldsymbol{\mu}))$ and $\mathbf{z}^2 = \sum_{v \in N^+(u)} \alpha_{u,v}^2 \psi^2(\mathbf{z}_1) + \mathbf{b}^2$ (which is just $\psi^2(\mathbf{z}_1) + \mathbf{b}^2$ since the weights sum to 1), then we have:

$$\begin{aligned}
|\mathbf{y}_u^2 - \mathbf{z}_2|_i = |\sum_{v \in N^+(u)} \alpha_{u,v}^2 [\psi^2(\mathbf{x}_v^1 - \mathbf{z}_1)]_i| &\leq \sum_{v \in N^+(u)} |\alpha_{u,v}^2 [\psi^2(\mathbf{x}_v^1 - \mathbf{z}_1)]_i| \\
&\leq \sum_{v \in N^+(u)} \alpha_{u,v}^2 \|\psi^2\|_\infty \|(\mathbf{x}_v^1 - \mathbf{z}_1)\|_\infty \\
&= \|\psi^2\|_\infty \|(\mathbf{x}_v^1 - \mathbf{z}_1)\|_\infty \\
&\leq \epsilon
\end{aligned}$$

where we used $\sum_{v \in N^+(u)} \alpha_{u,v}^2 = 1$ in the last step, which follows from the definition of the Dirichlet distribution. Thus for any $\epsilon > 0$ we have that for every node u and every $i = 1 \ldots d^2$

$$\mathbb{P}(|\mathbf{y}_u^2 - \mathbf{z}_2|_i \leq \epsilon) \to 1 \text{ as } n \to \infty$$

Now we inductively repeat this argument for an arbitrary number of layers.

Finally, once we apply the mean pooling layer, its output converges, because the final embedding of every node converges (this is trivial). Now it suffices to apply the non-splitting assumption, as was explained in the previous chapter. This concludes the proof. □

The above theorem proves the zero-one law for Dirichlet-GNN. Let us now look at some experiments.

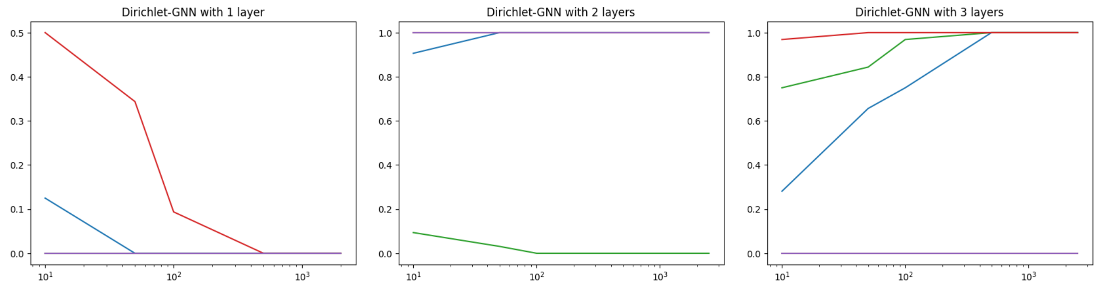
4.5 Empirical analysis of Dirichlet-GNN

Experimental setup Our experimental setup is quite similar to the setup in the paper [17]. We consider a Dirichlet-GNN with a varying number of layers. The model uses a truncated identity function as the non-linearity between the layers, and the weight matrices are initialized to come from a Xavier uniform distribution. For the final classifier, which is used to assign labels, we use a 2 layer MLP with $U(-1, 1)$ initialization.

The random graph model is the Erdős-Rényi model with $p = 1/2$. We consider graphs of sizes {10,50,100,500,1000,1500,2000,2500}. To estimate the probability of a label, we sample 32 graphs for each size. The initial node features are of are sampled from $U(0, 1)$, which is a sub-Gaussian distribution, and are of dimension 64, which is also the hidden dimension across the layers. The parameter in the Dirichlet distribution is set to be 1.

Results Figure 4.4 shows three plots, consisting of models with $1, 2$ and 3 layers. Each plot consists of 5 different models. The plots show nice convergence of all models, as is predicted by our theory.

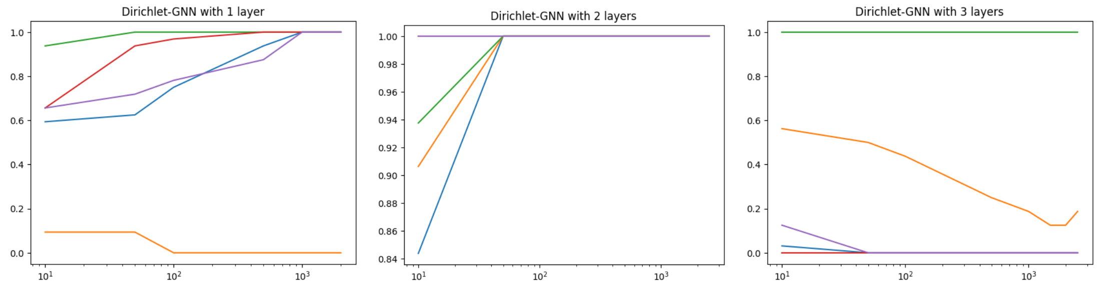
We can also experiment with the Dirichlet parameter q. Recall that the variance of each attention weight is inversely proportional to q. Thus if we decrease q, the



(a) 1-layer Dirichlet-GNN **(b)** 2-layer Dirichlet-GNN **(c)** 3-layer Dirichlet-GNN

Figure 4.4: Zero-one Law of Dirichlet-GNN with $q = 1$. The probability of a graph receiving the label 0, tends to 0 or 1, for every model.

attention mechanism should behave more wildly, and one would expect slower convergence. This is depicted in Figure 4.5. Note that in the 1-layer case, the convergence was slower, in the 2-layer case, the behaviour was similar as in the case of $q = 1$, and in the 3-layer case, interestingly, one model did not converge. However, it is important to keep in mind that the models are always initialized randomly, so it is difficult to make a fair comparison.



(a) 1-layer Dirichlet-GNN **(b)** 2-layer Dirichlet-GNN **(c)** 3-layer Dirichlet-GNN

Figure 4.5: Zero-one Law of Dirichlet-GNN with $q = 0.1$

4.6 Summary and discussion

In this chapter, we initiated our study of the zero-one law for attention networks. Inspired by common attention models, we defined Dirichlet-GNN, and studied its behavior. We began by exploring the convergence of Dirichlet random walks, and obtained results interesting from a purely mathematical point of view. In particular, we discovered some interesting properties of the Dirichlet distribution, such as

Theorem 4.2, Lemma 4.4 and Lemma 4.5, which allowed us to obtain the key bound in Theorem 4.7. Using the bound from Theorem 4.7, we theoretically proved that Dirichlet-GNN satisfies the zero-one law, and observed this behaviour in experiments.

Let us now discuss some consequences of this result. Even though Dirichlet-GNN is not a real graph machine learning model, the fact that it satisfies the zero-one law result does have interesting implications in this direction. In particular, the result suggests that many different MPGNNs are likely to satisfy the zero-one law, and that it is in fact quite difficult to design one that does not.

If we consider an MPGNN with attention values that are 'close' to $1/n$, where n is the number of nodes, and these attention weights sum up to 1, then the findings from this study suggest that such an MPGNN will probably satisfy the zero-one law. This requirement is not overly restrictive since most MPGNNs employ an attention mechanism that behaves similarly to this. The computation of a weighted mean is a natural choice for aggregating information, and by normalizing the weights to be close to $1/n$ and sum to 1, we ensure that the node features remain stable, preventing them from exploding or tending towards zero. Moreover, I believe that even if we relax the condition slightly, allowing for a deviation in the sum of the weights (as seen in the case of GCN), the zero-one law is still likely to hold.

Of course, these statements are somewhat informal, and we must exercise caution with such arguments. To claim definitively that a specific model satisfies the zero-one law, we cannot rely on Dirichlet-GNN; a separate, rigorous proof would be necessary. However, the insights gained from our current study provide a solid foundation for formulating an informed conjecture.

Definition 4.2

We say an attention weight $\alpha(n)$ is asymptotically close to a value $x(n)$, if $\mathbb{P}(|\alpha(n) - x(n)| > \epsilon) = O(1/n^2)$, for any $\epsilon > 0$.

The definition of closeness is motivated by the variance of the Dirichlet distribution, which decays as $1/n^2$, and Chebyshev's inequality.

Conjecture *Any MPGNN whose attention weights are asymptotically close to $1/N$, where N is the size of the neighborhood, converges, and satisfies a zero-one law with respect to the Erdős-Rényi random graph model.*

5

Extending the results beyond the ER model

So far, we have only talked about the zero-one law with respect to the random graph model $\mathbb{G}(n, p)$. It is completely natural to ask what happens for other random graph models. This is precisely what we will explore in this short chapter.

5.1 A slower Erdős-Rényi model

A natural way to tweak the $\mathbb{G}(n, p)$ is to decrease the probability of an edge being present as n increases, meaning that we consider p as a function of n, and write $p(n)$. We would like to understand the class of functions $p(n)$ under which we observe a zero-one law. Once again, this is motivated by a remarkable theorem in first-order logic. It turns out that random graph models $\mathbb{G}(n, n^{-a})$ where $a > 0$ is *irrational*, also satisfy a zero-one law. However, this result is well beyond the scope of this text and the interested reader is referred to [44].

Even though the above theorem states as a motivation, it is very unlikely that we will observe such a property in the zero-one laws of GNNs, in the sense that the rationality or irrationality of some parameter should not play a role. Intuitively, we know that the zero-one law effect occurs because of an averaging effect (recall our discussion about the laws of large numbers), and we simply need this averaging effect to be strong enough. Thus we will be interested in the asymptotic growth of $p(n)$. We have the following theorem.

Theorem 5.1
Let $p(n)$ be such that the following two conditions are satisfied

1. $\lim_{n\to\infty} np(n) = \infty$

2. $\lim_{n\to\infty} n\exp(-np(n)) = 0$

then under the same assumptions as in Theorem 4.1, the Dirichlet-GNN satisfies a zero-one law with respect to $\mathbb{G}(n, p(n))$

The first condition is obvious, because $np(n)$ is the expected value of the size of a neighborhood, and we want this to tend to infinity with n, while the second condition comes from the proof of Theorem 4.1. Note that we could in fact remove Condition 1, because it is implied by Condition 2.

This theorem gives us an easy way of checking if the law is satisfied. For example if $p(n) = \frac{\log(n)}{n^{\alpha}}$, where $\alpha < 1$, we have $\lim_{n\to\infty} n^{1-\alpha}\log(n) = \infty$, and $\lim_{n\to\infty} n\exp(-np(n)) = \lim_{n\to\infty} n\exp(-n^{1-\alpha}\log(n)) = \lim_{n\to\infty} n^{1-n^{1-\alpha}} = 0$, so by Theorem 5.2, we have a zero-one law. We can even remove the *log*, and simply consider $n^{-\alpha}$, and we see that the conditions are satisfied if $\alpha < 1$. Note that rationality/irrationality plays no role in the convergence behaviour.

Let's now prove the theorem.

Proof. The proof is identical to the proof of 4.1, only now the expected value of $|N(u)|$ is $np(n)$ instead of np where p is fixed. Thus the bound we obtain is

$$\begin{aligned}
&\mathbb{P}\left([\sum_{v\in N^+(u)} \alpha_{u,v}\psi^1(\mathbf{x}_v) - \psi^1(\boldsymbol{\mu})]_i > t\right) \\
&\quad\le \exp\left(\frac{-t^2}{\sigma^2}\frac{\frac{1}{2}np(n)\sqrt{\frac{1}{2}np(n)}q + \sqrt{\frac{1}{2}np(n)}}{q(\frac{1}{2}np(n) + \sqrt{\frac{1}{2}np(n)}) + \sqrt{\frac{1}{2}np(n)} + 1}\right) \\
&\quad + \frac{2\frac{1}{2}np(n)(\frac{1}{2}np(n) - 1)q(q+1)}{(\frac{1}{2}np(n)q + 1)^2(\frac{1}{2}np(n)q + 2)(\frac{1}{2}np(n)q + 3)} \\
&\quad + n\exp\left(-\frac{np(n)}{8}\right)
\end{aligned}$$

which tends to 0 under the assumptions of the theorem, and the rest of the proof is identical to that of Theorem 4.1. □

5.2 Stochastic block model

The stochastic block model (SBM) is a random graph model that generalizes the Erdős-Rényi model. In this model, we partition the vertex set into r sets $C_1, \dots, C_r$ called *communities*, and we have a symmetric $r \times r$ matrix P of probabilities. The vertices $v \in C_i$ and $u \in C_j$ are joined by an edge with probability P_{ij}. We will write $G \sim SBE(C_1 \dots C_r, P)$. Once again, we are interested in proving the zero-one law.

In the SBM, the distribution of the size of a node's neighborhood is a sum of r independent Binomial random variables with different parameter probabilities. Suppose $u \in C_i$. Then

$$|N^+(u)|= X_{i1} + X_{i2} + \cdots + X_{ir}$$

where $X_{ik} \sim Bin(|C_k|, p_{ik})$ represents the size of the neighborhood in cluster C_k. To make the proof of the zero-one law work, we must bound the probability that the size of the neighborhood of a node is not 'too small'. We can easily do this as follows. If we let $j = argmax_k |C_k| p_{i,k}$, then

$$\begin{aligned} \mathbb{P}(N^+(u) < \frac{|C_j|p_{i,j}}{2}) &\leq \mathbb{P}(X_{ij} < \frac{|C_j|p_{i,j}}{2}) \\ &\leq \exp(\frac{-|C_j|p_{i,j}}{8}) \end{aligned}$$

where we applied a Chernoff bound 2.10 in the last step.

The reason why we choose the argmax and not, say, the argmin, is that we do not want to require that all clusters be connected to each other (so some $p_{i,k}$ is allowed be zero) but we will only require that the 'largest' of the variables X_{ik} is 'large enough'.

For this reason, let $m_i = \max_k |C_k| p_{i,k}$ and $M = \min_i m_i$. Then for any node v we have

$$\mathbb{P}(N^+(v) < \frac{M}{2}) \leq \exp(\frac{-M}{8})$$

This allows us to only require that M grows enough. Note that that is equivalent to requiring that for any cluster C_i, there is a cluster C_j with enough edges (on average) between vertices in C_i and in C_j.

Using this bound, we can finish the proof of the zero-one law in the same way as in the proof of Thereom 4.1. Similarly as in the case of Theorem 5.2, we have the following theorem.

Theorem 5.2

Let $r \in \mathbb{N}$. Let $C_1(n) \ldots C_r(n)$ be a partition of n vertices and let $P(n)$ be an $r \times r$ matrix. Let $M(n) = \min_i(\max_k |C_k(n)| p_{i,k}(n))$, Suppose the following two conditions are satisfied

1. $\lim_{n\to\infty} M(n) = \infty$
2. $\lim_{n\to\infty} n \exp(-\frac{M(n)}{8})) = 0$

then under the same assumptions as in Theorem 4.1, the Dirichlet-GNN satisfies a zero-one law with respect to $SBE(C_1 \ldots C_k, P(n))$.

To summarize, the intuition behind the SBE model is quite obvious. If we require that for every cluster, there is some cluster with enough edges, the analysis is almost identical to that of the Erdős-Rényi model. However, note that this is a worst case analysis, and one could come up with different conditions on the partition function and probability matrix and still have convergence - one simply needs to ensure that each node has a high enough degree. The theorem serves to show that convergence under the SBE model is possible and relatively easy to obtain.

5.3 Barabási-Albert

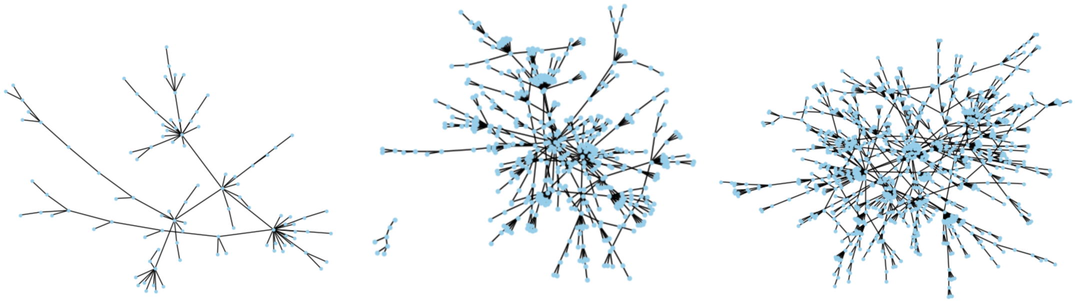
One issue with the Erdős-Rényi model is that it fails to model a class of graphs which often occurs in practice. It turns out, that many networks encountered in the real world contain collections of nodes with a high degree ('hubs') and many nodes with a low degree. This is common on social media, where most people have a low number of followers/friends, but a few well-known users have a very large number of followers. Furthermore, these well-known users have a higher probability that a new user will follow them rather than somebody who is not well-known. This is precisely the idea behind preferrential-attachment and the Barabási-Albert (BA) model.

In the BA model, the graphs are generated as follows. At each step, we add a new node v_i to the graph, and join v_i to a node u with probability proportional to $d(u)$. Formally, if p is the probability of an edge between u and v_i, we have

$$p = \frac{k}{\sum_j k_j}$$

where k is the degree of u and $\sum_j k_j$ is the sum of degrees over all pre-existing nodes. This ensures that at every step, nodes with a high degree have a higher probability of gaining a new edge than nodes with a lower one. A key difference between the BA model and ER model is that in the BA model, we build the graphs iteratively, whereas in the ER model the number of nodes is considered fixed. This is one reason why ER models are not good for modelling these kinds of networks.

It is quite obvious to see that our proof of the zero-one law does not extend to this class of graphs. Recall that in the proof, we require all node degrees to be 'large', but in the BA model, this is not the case almost by definition. The preferential-attachment algorithm ensures that there are plenty of nodes with a small degree. In fact, one can show that the degree distribution follows a power-law distribution, where the probability that a node has degree k is proportional to k^{-3}.

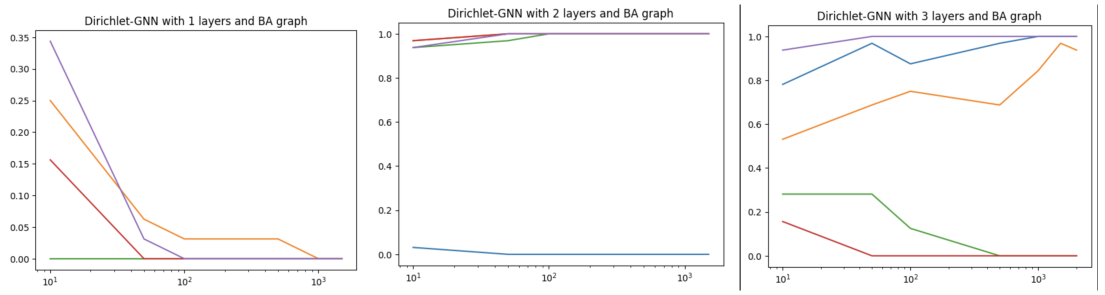


(a) BA graph with 100 nodes **(b)** BA graph with 1000 nodes **(c)** BA graph with 1500 nodes

Figure 5.1: Examples of BA graphs

Let us look at some experiments. We will use the same experimental setup as in the previous section, but now, we will generate Barabási-Albert graphs. We will do so by only adding 1 edge at each step of the algorithm, which will ensure there will be many nodes with a low degree, and the zero-one law should not be satisfied. Examples of the graphs we generate are shown in Figure 5.1.

Interestingly, if we use the same experimental setup as in the previous section, the plots show a similar behaviour as in the case of a zero-one law, as can be seen in Figure 5.2. How is this possible?



(a) 1-layer Dirichlet-GNN **(b)** 2-layer Dirichlet-GNN **(c)** 3-layer Dirichlet-GNN

Figure 5.2: Dirichlet-GNN with BA graphs

It is difficult to answer this question exactly, but most likely, this is caused because the final mean-pooling layer is ‘too strong’. The mean pooling layer probably causes the output to be concentrated around some value even if the individual node features nor the mean actually converge. This means that in order to not see the zero-one law in experiments, we would need to choose a classifier such that the classification boundary of the final classifier is close to this value, in which case we would see oscillatory behaviour of the labels. But designing such experiments and concluding that the zero-one law is not satisfied is not fair. One could setup a classifier with the classification boundary close to the mean in cases where convergence does occur, and obtain plots that look suggest the zero-one law is not satisfied, when in fact it would occur for much larger graph sizes.

This suggest that perhaps we are not paying attention to the right thing. After all, our theory suggests something more than just the zero-one law. In particular, we know that the embeddings of each node converge, and the zero-one law is just a consequence of this fact. Thus a better approach would be to check whether the node embeddings converge. In the Barabasi Albert model, this is obviously not the case for nodes with a low degree.

This is the approach that we will use in the next part of the thesis. Instead of performing experiments which check the zero-one law, we will focus on the actual limit of the embeddings. Naturally, the zero-one law is still our primary focus because of the implied expressiveness limitations, but it will be a consequence of the convergence.

6
Extending the zero-one law to further models

In the previous section, we began our discussion of the convergence behaviour of attention networks, and we proved the zero-one law for Dirichlet-GNN. However, as was mentioned before, Dirichlet-GNN is not an actual model, but merely a tool we used for understanding the general behaviour of attention networks. In this section, we will consider three concrete attention models, and explore their convergence behaviour. We will show that under some conditions, convergence occurs, and we will even be able to explicitly compute this limit.

Unfortunately, to do this, we cannot use results from the previous section, but we must come up with completely different techniques. The core idea in our approach is strongly inspired by a very recent paper [18], which builds on previous results in this area [45, 46]. In this section, we will first analyze and carefully explain the approach presented in [18].

Once we gain an understanding of this novel approach, we will extend the theory to cover the case of three concrete attention models, which is an original contribution of this thesis, because the paper itself *does not prove convergence for any model.* Finally, we will relate our results to the zero-one law. Throughout this chapter, we will use the same notation as is used in the paper [18].

6.1 Problem statement, setup, and notation

Our setup will be a little different from the previous section. Firstly, we will begin by only considering fully connected graphs, and only extend the results to E-R graphs later. Secondly, the node features will not be sub-Gaussian, but we place a stronger restriction - we require the variables to be bounded. This is a stronger restriction because one can show that any bounded random variable is sub-Gaussian.

For the notation, we let $(X_i)_{i\in\mathbb{N}}$ be a sequence of independent and identically distributed random variables, taking values in a subspace $X \subset \mathbb{R}^d$. These random variables represent the initial node features. We denote the probability measure, which produces the distribution of X_i by P. We place no restriction on the distribution of X_1, but we require that X is compact. Recall that by the Heine-Borel theorem, this is equivalent to requiring X to be closed and bounded. We will often use the basic fact from analysis, that the image of a continuous map acting on a compact set is compact. This will allow us to obtain bounds in some steps.

In this chapter, we will always assume that a layer in an attention network uses the following aggregation scheme

$$z_i^{l+1} = \sum_{j\in N(i)}^{n} \frac{c^l(z_i^l, z_j^l)}{\sum_{k=1}^{n} c^l(z_i^l, z_k^l)} \psi^l(z_j^l)$$

where z_i^l are the node features given by the previous layers, c^l is some continuous function representing the attention mechanism and ψ a linear transformation. Transformers (2.8), GATs (2.9), GATv2 (2.10) fall into this category, but GCN (2.4), MEANGNN (2.5), DirichletGNN (4.1) do not.

The question that we would like to answer is: What happens to the output of these models as the number of nodes in the (fully-connected) graph goes to infinity? The following section presents an approach which will answer this question. We will begin by focusing on Transformers and consider GAT, GATv2 later.

6.2 First steps

In previous sections, we saw that by using the laws of large numbers, we could obtain important insights into the convergence behaviour of the model, and actually ‘guess’ what the limit should be. After doing so, we were able to obtain a bound on the probability that the actual output is far from this mean, and show it goes to zero.

If we attempt to do the same in the case of attention models, we quickly run into difficulties. To illustrate this, consider the first layer, where the initial node features X_i are independent random variables. However, even then, the variables $c(X_i, X_j)\psi^l(X_j)$ are not independent, because every term is a function of X_i. This means we cannot apply the law of large numbers, and it is not obvious whether convergence occurs, and if it does, what the limit is.

The main idea of the approach we will shortly present is as follows. If we condition on the random variable X_i, we could treat $c(X_i, X_j)\psi^l(X_j)$ as independent. In that case, we could divide both the numerator and denominator by $1/n$ (which doesn’t change anything), and apply the law of large numbers, to both terms. Then, we could use the law of total probability to get rid of the conditioning. Then we will utilize a concentration bound to obtain a convergence result. Although it may not sound that complicated, making this idea work rigorously requires a lot of technical results. It is important you keep this idea in mind as you read the theory in the following sections.

A note for the markers: Throughout the next sections, we will often use definitions, theorems and proofs from [18], and these will always be cited.

6.3 Theoretical analysis of a single layer

Let us now begin a formal analysis of the problem. We have the following definitions

Definition 6.1 ([18]).

The $l-th$ layer of our attention network will be denoted by the function F^l, where

$$F^l(z_i^l, \{\{z_j^l\}\}_{v_j \in N(v_i)}) = \sum_{j \in N(i)}^{n} \frac{c^l(z_i^l, z_j^l)}{\sum_{k=1}^{n} c^l(z_i^l, z_k^l)} \psi^l(z_j^l) = z_i^{l+1}$$

In particular, throughout this section, we will assume there are no non-linearities between the layers. This is something we will deal with later.

Definition 6.2 ([18]).
The node features after the $l-th$ layer are a matrix of size $n \times d_l$, denoted by S_X. We write $(S_X)_i$ to represent the node features around node i. Note that (S_X) is a random variable.

Now we introduce a crucial concept.

Definition 6.3 ([18]).
Let $x \in X$. For $l \geq 1$, we recursively define

$$f^{l+1}(x) = \int_{y \in X} \frac{c^l(f^l(x), f^l(y))}{\int_{t \in X} c^l(f^l(x), f^l(t)) dP(t)} \psi^l(f^l(y)) dP(y)$$

and $f^0(x) = x$. We call f^l the l-th ***continuous counterpart****.*

It is essential that we understand the above definition. In particular, note that continuous counterparts are ordinary, deterministic functions, because the x is fixed and the ‘randomness’ is integrated out. Another way of writing it is:

$$f^{l+1}(x) = \frac{\mathbb{E}\left[c^l(f^l(x), f^l(X_1))\psi^l(f^l(X_1)))\right]}{\mathbb{E}\left[c^l(f^l(x), f^l(X_1))\right]}$$

Let us now explore the properties of continuous counterparts. We first need some standard definitions. We denote the sup norm of a vector $x \in \mathbb{R}^d$ by $\|\cdot\|_\infty$, where $\|x\|_\infty = \max_{i=1,\dots d}|x_i|$. For a function $f : X \to \mathbb{R}^d$, we will denote by $\|f\|_\infty = \sup_{x \in X}\{\|f(x)\|_\infty\}$. For a linear map, we define

Definition 6.4
Let $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ be normed spaces. Let $\psi : X \to Y$ be linear. Then we define

$$\|\psi\| = \sup_{x \in X, \|x\|_X \neq 0} \frac{\|\psi(x)\|_X}{\|x\|_Y}$$

We say ψ is bounded if $\|\psi\| < \infty$.

We will sometimes specify the choice of the norm by using special notation. For example, the notation $\|\psi\|_2$ means that the norm $\|\cdot\|_2$ is acting both on X and Y (in fact these are two different norms, but the idea is clear). Finally, note that if ψ is bounded, then for any $x \in X$ we have $\|\psi(x)\|_Y \leq \|\psi\|\|x\|_X$.

Let us now examine some properties of the continuous counterpart in the case of transformer models.

Lemma 6.1

If f^l is bounded, then $\exists \alpha^l, \beta^l > 0$ such that $\forall x, y$ we have $\alpha^l \leq c^l(f^l(x), f^l(y)) \leq \beta^l$, and f^{l+1} is bounded.

Proof. (In the transformer case)

Recall that by definition of c^l, we have

$$c^l(f^l(x), f^l(y)) = \exp \frac{(\langle \psi_1^l(f^l(x)), \psi_2^l(f^l(y)\rangle)}{\sqrt{d^l}}$$

Now, by the Cauchy-Schwarz inequality 2.14, we know that

$$\begin{aligned}
|\langle \psi_1^l(f^l(x)), \psi_2^l(f^l(y)\rangle| &\leq \|\psi_1^l(f^l(x))\|_2 \|\psi_2^l(f^l(y))\|_2 \\
&\leq \|\psi_1^l\|_2 \|f^l(x)\|_2 \|\psi_2^l\|_2 \|f^l(y)\|_2 \\
&\leq \|\psi_1^l\|_2 \|\psi_2^l\|_2 d_{l+1} \|f^l\|_\infty^2 \\
&= C
\end{aligned}$$

where d_l, d_{l+1} are the dimension of the inputs and image of f^l respectively, and $\|f^l\|_\infty < \infty$ by assumption. Note that we used Lemmas 2.16 and 2.15. Now, since the absolute value of the inner product is bounded, once we pass it to the exponential function, the result is bounded from above and from below. Since $exp(\cdot)$ is strictly positive, we have that $\exists \alpha^l, \beta^l > 0$ such that

$$\alpha^l \leq c^l(f^l(x), f^l(y)) \leq \beta^l$$

as required.

It now suffices to show that f^{l+1} is bounded. But from the definion of f^{l+1}, we have that

$$\begin{aligned}\|f^{l+1}\|_\infty &\leq \frac{\beta^l \|\psi\|_\infty \|f^l\|_\infty}{\alpha^l} \\ &\leq \infty\end{aligned}$$

as desired. □

We can now use this lemma to obtain the following

Lemma 6.2

Suppose we have L layers. Then for all $l = \{0, 1 \ldots L\}$, f^l is bounded. Furthermore, there exists $\alpha, \beta > 0$ such that $\forall l$ and $\forall x, y$, $\alpha < c^l(x, y) < \beta$

Proof. It suffices to show f^0 is bounded and then use induction combined with the previous lemma. This is trivial because $f^0(x) = x$ and X is assumed to be compact and thus bounded. Thus $\|f^0\|_\infty < \infty$. Now applying the previous lemma for every layer and setting $\alpha = \min_l \alpha^l$ and $\beta = \max_l \beta^l$, we obtain the result. □

One key result that we will need is the fact that if X_i and X_j are independent, then $f(X_i)$ and $f(X_j)$ are independent. This fact is true provided that f is a *measurable* function. Even though this fact is quite crucial for us, I've decided not to include the formal definition of measurability, nor the proof of this fact, because this section is already quite technical. The interested reader can quickly verify that this is the case, and the idea of the proof is illustrated below.

Lemma 6.3

For $l \geq 0$, f^l is measurable.

Proof. Follows from Lemma 6.2, Fubini's Theorem, the fact that ratio of measurable functions is measurable, the fact that $g \circ f$ is measurable if g is continuous and f is measurable, and finally, induction. □

Given these definitions and results, we are now ready to prove the first convergence results for transformers. However, before we dive into the technicalities, it is worth explaining the main ideas in simpler terms. In particular, the question remains why we defined f^l in this way in the first place, why are continuous counterparts interesting and how are they even related to the problem we are studying.

The reason is that it turns out that the every layer of our GNN in fact converges to f^l. More precisely, we will show that $\max_i \|(S_X)_i^l - f^l(X_i)\|_\infty$ converges to 0 in distribution. How can we see that? Suppose we fix $X_i = x_i \in X$ and consider

$$\|(S_X)_i^1 - f^1(x_i)\|_\infty = \|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - f^1(x_i)\|_\infty$$

Note that we are fixing the random variable X_i, but keeping the remaining features random. This allows us to do a very clever trick - the term $F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)})$ is a random variable as a function of X_j, $j \neq i$ and therefore we can consider its expectation. The idea is that perhaps, if we are lucky, we could use some kind of concentration bound to show that this random variable is tightly concentrated around its expectation. With this idea in mind, define

$$F_n(x_i) = \mathbb{E}_{X_j, j \neq i}\left[F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)})\right]$$

and now use the triangle inequality, to get

$$\|(S_X)_i^1 - f^1(x_i)\|_\infty \leq \|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - F_n(x_i)\|_\infty + \|F_n(x_i) - f^1(x_i)\|_\infty$$

If it were the case that F^1 is tightly concentrated around its expectation, we could obtain a bound for the first term on the right-hand-side. But what about the second term? It turns out, that $\|F_n(x_i) - f^1(x_i)\|_\infty \to 0$ as $n \to \infty$. As we will see shortly, this observation is actually very intuitive, and this is the reason why f^1 is defined the way it is.

This means that we will need to prove two things. Firstly we need to show that F^1 is tightly concentrated around its expectation. This will be handled by the McDiarmid inequality 2.11. Secondly, we need to show $\|F_n(x_i) - f^1(x_i)\|_\infty \to 0$. It turns out that the first claim is quite technical, and for this reason, we begin by proving the second claim.

Note that the following definition and theorem are stated and proved for an arbitrary number of layers.

Definition 6.5 ([18]).
Let $l \geq 0$. We define $F_n^{l+1}(f^l)(\cdot) : X \to \mathbb{R}^{d_{l+1}}$ by

$$F_n^{l+1}(f^l)(x_i) = \mathbb{E}_{X_j, j \neq i}\left[F^{l+1}(f^l(x_i), \{\{f^l(X_j)\}\}_{v_j \in N(v_i)})\right]$$

Theorem 6.4 ([18]).
$\|F_n^{l+1}(f^l)(x_i) - f^{l+1}(x_i)\|_\infty \to 0$ *as* $n \to \infty$.

Proof. This proof is quite different from the one presented in the paper, and hopefully more intuitive.

Write

$$\begin{aligned} F^{l+1}(f^l(x_i), \{\{f^l(X_j)\}\}_{v_j \in N(v_i)}) &= \sum_{j \in N(v_i)}^{n} \frac{c^{l+1}(f^l(x_i), f^l(X_j))}{\sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(X_k))} \psi^{l+1}(f^l(X_j)) \\ &= \frac{1/n \sum_{j \in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(X_j)) \psi^{l+1}(f^l(X_j))}{1/n \sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(X_k))} \end{aligned}$$

For clarity, let

$$\begin{aligned} Z_j &= c^{l+1}(f^l(x_i), f^l(X_j)) \psi^l(f^l(X_j)) \\ Y_k &= c^{l+1}(f^l(x_i), f^l(X_k)) \end{aligned}$$

and the above becomes

$$= \frac{1/n \sum_{j \in N(v_i)}^{n} Z_j}{1/n \sum_{k \in N(v_i)}^{n} Y_k}$$

Since the random variables X_m are i.i.d and by Lemma 6.3, we have that f^l is a measurable function, $f^l(X_m)$ are also i.i.d, and thus Z_j are i.i.d and Y_k are i.i.d. This means we are summing independent and identically distributed random variables. Note that by lemma 6.1, both Z_j and Y_k are bounded and thus integrable. Furthermore, Y_k is bounded away from zero. Thus we can use the strong law of large numbers and the algebra of limits, to conclude that

$$\frac{1/n \sum_{j \in N(v_i)}^{n} Z_j}{1/n \sum_{k \in N(v_i)}^{n} Y_k} \to \frac{\mathbb{E}[Z_1]}{\mathbb{E}[Y_1]} \text{ a.s}$$

This convergence is dominated, because

$$\left\|\frac{1/n\sum_{j\in N(v_i)}^n Z_j}{1/n\sum_{k\in N(v_i)}^n Y_k}\right\|_\infty \leq \frac{\beta\|\psi\|_\infty\|f^l(X_j)\|_\infty}{\alpha}$$

and clearly $\int \frac{\beta\|\psi\|_\infty\|f^l(X_j)\|_\infty}{\alpha}dP = \frac{\beta\|\psi\|_\infty\|f^l(X_j)\|_\infty}{\alpha} < \infty$. Thus by the dominated convergence theorem, we have that

$$\mathbb{E}_{X_j j\neq i}\left[\frac{1/n\sum_{j\in N(v_i)}^n Z_j}{1/n\sum_{k\in N(v_i)}^n Y_k}\right] \to \mathbb{E}\left[\frac{\mathbb{E}[Z_1]}{\mathbb{E}[Y_1]}\right] = \frac{\mathbb{E}[Z_1]}{\mathbb{E}[Y_1]}$$

Now it remains to observe observe that the left-hand side is precisely $F_n^{l+1}(f^l)(x_i)$, while the right-hand side is precisely $f^{l+1}(x_i)$:

$$\begin{aligned}\frac{\mathbb{E}[Z_1]}{\mathbb{E}[Y_1]} &= \frac{\mathbb{E}[c^l(f^l(x_i), f^l(X_1))\psi^l(f^l(X_j))]}{\mathbb{E}[c^l(f^l(x_i), f^l(X_1)]}\\ &= \int_{y\in X}\frac{c^l(f^l(x), f^l(y))}{\int_{t\in X} c^l(f^l(x_i), f^l(t))dP(t)}\psi^l(f^l(y))dP(y) = f^{l+1}(x_i)\end{aligned}$$

thus $F_n^{l+1}(x_i) \to f^{l+1}(x_i)$ as $n \to \infty$, as required. □

In the original paper, a different approach was used to prove this claim. Our method is more insightful, as it clearly demonstrates the origin of the continuous counterpart's definition and explains the necessity for the conditioning. However, the proof presented in the paper actually provides an upper bound on how quickly this convergence happens, which is nice, but not relevant for our purposes.

Having proved that the expectation of each layer converges pointwise to f^{l+1}, we can shift our focus to the other term. We need to bound $\|F^1(x_i, \{\{X_j\}\}_{v_j\in N(v_i)}) - F_n(x_i)\|_\infty$. Observe that we can treat $F^1(x_i, \{\{X_j\}\}_{v_j\in N(v_i)})$ as a function of the $n-1$ independent random variables $\{X_j\}_{j\neq i}$. If we denote by Z_i the tuple $(X_1, \dots X_n)$ where X_i is omitted, we can denote $F^1(x_i, \{\{X_j\}\}_{v_j\in N(v_i)})$ by $g(x_i)(Z_i)$. Since $g(x_i)(\cdot)$ is a function of independent random variables, we can use McDiarmid's inequality 2.11 to bound the distance from its mean with high probability. In order to apply this theorem, we need to verify the bounded differences property, which is the purpose of the following definition and theorem.

Definition 6.6

Let $l \geq 0$. We will denote by $g_n^{l+1}(x_i)(\cdot)$: $X^{n-1} \to \mathbb{R}^{d_{l+1}}$ defined by $g_n^{l+1}(x_i)(Z_i) = F^{l+1}(f^l(x_i), \{\{f^l(X_j)\}\}_{v_j\in N(v_i)})$, where $Z_i = (X_1 \dots X_n)$ with X_i omitted.

Theorem 6.5 ([18]).
Let $l \in \{0, \dots L\}$, $i \in \{1, \dots n\}$, $x_i \in X$. Let $m \in \{1, \dots, n\}$ with $m \neq i$. Fix $X_j = x_j$ for $j \neq i$ and let $z = (x_1 \dots x_n)$ with x_i omitted. Let $x'_m \in X$ and $x'_j = x_j$ for $j \neq k$. Let $z' = (x'_1 \dots x'_n)$ with x'_i omitted. Then

$$\|g_n^{l+1}(x_i)(z) - g_n^{l+1}(x_i)(z')\|_\infty \leq D_n^l$$

and $D_n^l = O(1/n)$.

[18]. The proof is very technical. Please see Appendix A. □

Given this result, we can finally apply McDiarmid's inequality to obtain a concentration bound in the first layer, and this is the purpose of the following theorem. We illustrate its proof because in later sections, we will need to modify it.

Theorem 6.6 ([18]).
Let $a_n = \|F_n^1(f^0)(x_i) - f^1(x_i)\|_\infty$ and consider $\rho \in (0, 1]$. Then

$$\mathbb{P}\left(\|(S_X)_i^1 - f^1(x_i)\|_\infty \leq D_n^1 \sqrt{\frac{n-1}{2} \ln \frac{2d_1 n}{\rho}} + a_n\right) \geq 1 - \frac{\rho}{n}$$

Proof. [18] Fix $X_i = x_1$, We have:

$$\begin{aligned}\|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - f^1(x_i)\|_\infty &\leq \|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - F_n(x_i)\|_\infty + \|F_n(x_i) - f^1(x_i)\|_\infty \\ &= \|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - F_n(x_i)\|_\infty + a_n\end{aligned}$$

Applying McDiarmid's inequality to the first term, and noting that $\sum_i c_i^2 \leq (n-1)D_n^2$, we obtain

$$\mathbb{P}\left(\|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - f^1(x_i)\|_\infty \leq D_n^1 \sqrt{\frac{n-1}{2} \ln \frac{2d_1 n}{\rho}} + a_n\right) \geq 1 - \frac{\rho}{n}$$

Now we can simply use the law of total probability. For clarity, define $C_n = D_n^1 \sqrt{\frac{n-1}{2} \ln \frac{2d_1 n}{\rho}}$. We have:

$$\begin{aligned}\mathbb{P}\left(\|(S_X)_i^1 - f^1(x_i)\|_\infty \leq C_n + a_n\right) &= \int \mathbb{P}\left(\|(S_X)_i^1 - f^1(x_i)\|_\infty \leq C_n + a_n | X_i = x_i\right) dP \\ &\geq \int 1 - \frac{\rho}{n} dP = 1 - \frac{\rho}{n}\end{aligned}$$

□

The above result only applies to a fixed node. We would like to ensure that a concentration bound holds for all nodes simultaneously. This is described in the following result.

Corollary 6.1 ([18]).

$$\mathbb{P}\left(\max_i \|(S_X)^1_i - f^1(X_i)\|_\infty \leq D_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}} + a_n\right) \geq 1-\rho$$

Proof. [18] A simple application of the union bound. □

We can now extend these result to cover the case of the zero-one law, in the one-layer case.

Corollary 6.2

$\max_i\|(S_X)^1_i - f^1(X_i)\|_\infty$ *converges to 0 in distribution, as* $n \to \infty$.

Proof. Since $a_n \to 0$ as $n \to \infty$ and since $D_n = O(1/n)$, we have for arbitrary $\rho > 0$ that:

$$\lim_{n\to\infty} D_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}} + a_n = 0$$

Now fix any $x > 0$ and suppose $\epsilon > 0$ is given. Letting $\rho = \epsilon$ and taking n large enough so that $D_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}} + a_n < x$, we have that $\mathbb{P}\left(\max_i\|(S_X)^1_i - f^1(X_i)\|_\infty > x\right) < \epsilon$. Thus, by the definition of a limit

$$\lim_{n\to\infty}\mathbb{P}\left(\max_i\|(S_X)^1_i - f^1(X_i)\|_\infty > x\right) = 0$$

which is exactly convergence in distribution to the random variable 0. □

Now suppose that we have a one-layer transformer and we use it to classify graphs. To do this, we use mean aggregation after the first layer, so we compute $\frac{1}{n}\sum_{i=1}^n (S_X)^1_i$. Using the previous result, we obtain a result which is (almost) the zero-one law.

Lemma 6.7

Let $\mu = \mathbb{E}[f^1(X_1)]$. *Then* $\frac{1}{n}\sum_{i=1}^n (S_X)^1_i$ *converges to* μ *in distribution.*

Proof. We have

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n}(S_X)_i^1-\mu\right\|_\infty &\leq \left\|\frac{1}{n}\sum_{i=1}^{n}(S_X)_i^1-\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)\right\|_\infty+\left\|\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)-\mu\right\|_\infty \\ &\leq \max_i\|(S_X)_i^1-f^1(X_i)\|_\infty+\left\|\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)-\mu\right\|_\infty\end{aligned}$$

Now suppose $x>0$ is given. Then

$$\begin{aligned}&\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i=1}^{n}(S_X)_i^1-\mu\right\|_\infty>x\right)\\ &\leq \mathbb{P}\left(\max_i\|(S_X)_i^1-f^1(X_i)\|_\infty+\left\|\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)-\mu\right\|_\infty>x\right)\\ &\leq \mathbb{P}\left(\max_i\|(S_X)_i^1-f^1(X_i)\|_\infty>x/2\bigvee\left\|\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)-\mu\right\|_\infty>x/2\right)\\ &\leq \mathbb{P}\left(\max_i\|(S_X)_i^1-f^1(X_i)\|_\infty>x/2\right)+\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)-\mu\right\|_\infty>x/2\right)\end{aligned}$$

By the strong law of large numbers 2.8, we know that

$$\frac{1}{n}\sum_{i=1}^{n}f^1(X_i)\to\mu\text{ a.s}$$

and by Lemma 2.7 we know that almost sure convergence implies convergence in probability. Thus the limit of the second term is 0. By the previous corollary, we know that the first term converges to 0. Thus taking limits, we have

$$\lim_{n\to\infty}\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i=1}^{n}(S_X)_i^1-\mu\right\|_\infty>x\right)\leq 0+0=0$$

Since x was arbitrary, the claim follows. □

Now we are almost finished. Recall that in order to classify the graphs, we would introduce some kind of a binary classifier C after the final layer. Similarly as in the case of GCN, we need to place a restriction on the classifier. Recall that for GCN/MeanGNN we required the classifier to be *non-splitting* (Definition 3.6. We can define a non-splitting classifier analogously for a general MPGNN:

Definition 6.7 (Non-splitting classifier for a general MPGNN).
Let M be an MPGNN equipped with a mean pooling layer. Let G_n be a sequence of graphs sampled from a random graph model, and let $\mathbf{F}_n$ be a matrix of node features

sampled from a distribution $\mathbb{D}$. Suppose that the output of M converges to a limit L in distribution. Then, we say a binary classifier C is non-splitting for M, if L does not lie on the decision boundary of C.

The zero-one law then follows from Theorem 2.9.

Theorem 6.8

*Let $X \subset \mathbb{R}^d$ be compact, let P be a probability measure on X, and let (X_n) be a sequence of independent random variables with distribution given by P. Suppose there exists $\alpha, \beta > 0$ such that $\alpha \leq c(x, y) \leq \beta \; \forall x, y \in X$. Then any **one-layer** attention network with c acting as the attention mechanism, equipped with a mean-pooling layer and a non-splitting classifier, satisfies the zero-one law with respect to (X_n) as the initial node features, on fully-connected graphs.*

In particular, note that in view of Lemma 6.1, the Transformer (2.8) satisfies the assumptions of Theorem 6.8, and therefore it satisfies the zero-one law.

However, note that this is a non-standard way to formulate the zero-one law. Previously, we were sampling graphs from the Erdős-Rényi model. Now we are only considering fully connected graphs (for the time being), which means the probability is only taken with respect to the node features, not with respect to the sampled graph (because the graph is deterministic).

6.4 Experiments for a single layer of a Transformer

Before we cover the case of multiple layers, let us experimentally verify the convergence in the one-layer case. As we will shortly see, designing experiments which validate the claims presented in the previous section is not that straightforward, and for this reason, we first focus on only one layer.

In view of our discussion in Chapter 5, we know that experiments that only consider the zero-one law are not enough. Luckily, the presented theoretical framework gives us an exact limit to which the models should converge, and the zero-one law is a consequence of this. Thus, it is a better idea to look at the actual node embeddings, instead of the zero-one law.

In order to do this, we must compute the theoretical limit to which our models converge. For example, Lemma 6.7 tells us that this limit is $\mathbb{E}[f(X)]$. Thus we need to analytically compute the continuous counterpart and its expectation and compare its value to the output of the GNN. The question is, how do we find the continuous counterpart?

We know, by definition, that in the one layer case, the continuous counterpart is

$$f(x) = \int_{y \in X} \frac{c(x,y)}{\int_{t \in X} c(x,t)dP(t)} \psi(y)dP(y)$$

Firstly, note that f depends on the underlying probability measure P, meaning that as we sample the features from different distributions, we get different continuous counterparts - as one would expect. Secondly, f is defined in terms of two integrals, the analytical computation of which can be difficult, or even impossible, depending on which probability measure P we choose. Furthermore, in light of Lemma 6.7, we would like to compute the *expectation* of $f(X)$, which means we will have to further integrate this with respect to $dP(x)$.

For most distributions on X (i.e most measures P), this computation is hard. For this reason, we will begin with a very simple choice for P, namely the Lebesgue measure on $[0,1]^d$ which is equivalent to the Uniform($[0,1]^d$) distribution. However, already, the computation of f, and especially its expectation, will prove to be quite challenging. We have the following result:

Theorem 6.9

Let $\psi, \psi_1, \psi_2 \in \mathbb{R}^{d_1 \times d}$ be the weight matrices in a one-layer transformer. Let $x \in [0,1]^d$ be arbitrary and let $c = \frac{1}{\sqrt{d}}\psi_2^T \psi_1 x$. Then the continuous counterpart (Definition 6.3) with respect to the $U[0,1]^d$ distribution, is given by

$$f(x) = \psi z$$

where

$$z_i = \frac{e^{c_i}(c_i - 1) + 1}{c_i(e^{c_i} - 1)}$$

To prove this theorem, we first need two lemmas.

Lemma 6.10

Let $d \in \mathbb{R}$ and $c_i \neq 0$ for $i = 1, \dots d$. Let

$$I_d = \int_0^1 \int_0^1 \dots \int_0^1 \exp(c_1 x_1 + c_2 x_2 + \dots c_d x_d) dx_1 dx_2 \dots dx_d$$

Then $I_d = \prod_{i=1}^d \frac{e^{c_i} - 1}{c_i}$

Proof. By induction. For $d = 1$ we have:

$$I_1 = \int_0^1 \exp(c_1 x_1) dx_1 = \left[\frac{1}{c_1} \exp(c_1 x_1) \right]_0^1 = \frac{\exp c_1 - 1}{c_1}$$

Now suppose the formula is true for d. Then $I_{d+1} = I_d \int_0^1 \exp(c_{d+1} x_{d+1}) dx_{d+1} = \prod_{i=1}^{d+1} \frac{e^{c_i} - 1}{c_i}$, as required. □

Lemma 6.11

Let $d \in \mathbb{N}$, $c_i \neq 0$ for $i = 1, \dots d$ and $w \neq 0$. Let

$$I_d = \int_0^1 \int_0^1 \dots \int_0^1 \exp(c_1 x_1 + c_2 x_2 + \dots c_d x_d) w x_1 dx_1 dx_2 \dots dx_d$$

Then $I_d = w \frac{e^{c_1}(c_1 - 1) + 1}{c_1^2} \prod_{j=2}^d \frac{e^{c_j} - 1}{c_j}$

Proof. We have:

$$\begin{aligned} I_d &= w \int_0^1 \int_0^1 \dots \int_0^1 \exp(c_2 x_2 + \dots c_d x_d) \exp(c_1 x_1) x_1 dx_2 \dots dx_d \\ &= w \prod_{j=2}^d \frac{e^{c_j - 1}}{c_j} \int_0^1 \exp(c_1 x_1) x_1 dx_2 dx_2 \dots dx_d \end{aligned}$$

Integrating by parts, we have:

$$\begin{aligned} &= w \prod_{j=2}^d \frac{e^{c_j - 1}}{c_j} \left(\frac{e^{x_1} x_1}{c_1} |_0^1 - \frac{1}{c_1} \int_0^1 \exp(c_1 x_1) \right) \\ &= w \prod_{j=2}^d \frac{e^{c_j - 1}}{c_j} \frac{e^{c_1}(c_1 - 1) + 1}{c_1^2} \end{aligned}$$

□

We can now prove Theorem 6.9.

Proof Of Theorem 6.9. By definition, we know

$$f(x) = \frac{\int_0^1 \exp(\frac{1}{\sqrt{d}}\langle \psi_1 x, \psi_2 y\rangle)\psi y \, dy}{\int_0^1 \exp(\frac{1}{\sqrt{d}}\langle \psi_1 x, \psi_2 y\rangle)dt}$$

But note that $\frac{1}{\sqrt{d}}\langle \psi_1 x, \psi_2 y\rangle = c_1 y_1 + \dots c_d y_d$, where $c_j = \frac{1}{\sqrt{d}}\langle (\psi_2)_j, \psi_1 x\rangle$, where $(\psi_2)_j$ denotes the i column of ψ_1. Note that c_j is in fact a function of x. We will denote by c the vector with components c_j, and we have $c = \frac{1}{\sqrt{d}}\psi_2^T \psi_1 x$. Thus we can rewrite f as:

$$f(x) = \frac{\int_0^1 \exp(c_1 y_1 + c_2 y_2 \dots c_d y_d)\psi y \, dy}{\int_0^1 \exp(c_1 t_1 + c_2 t_2 \dots c_d t_d)dt}$$

If we let z be the vector where

$$z_i = \frac{\int_0^1 \exp(c_1 y_1 + c_2 y_2 \dots c_d y_d) y_i \, dy}{\int_0^1 \exp(c_1 t_1 + c_2 t_2 \dots c_d t_d)dt}$$

Then by linearity of the integral, we have

$$f(x) = \psi z$$

Once again, note that $z = z(x)$ is a function of x. Using the previous Lemma 6.10 and Lemma 6.11, we have:

$$\begin{aligned} z_i &= \frac{\frac{e^{c_i}(c_i - 1) + 1}{c_i^2} \prod_{j \neq i}^d \frac{e^{c_j} - 1}{c_j}}{\prod_{i=1}^d \frac{e^{c_i} - 1}{c_i}} \\ &= \frac{e^{c_i}(c_i - 1) + 1}{c_i(e^{c_i} - 1)} \end{aligned}$$

which gives us the desired closed form expression for $f(x)$. □

In view of Lemma 6.7, we also need to compute

$$\int_0^1 f(x) dP(x)$$

However, as we will now demonstrate, there is no closed form solution to this integral. Let us explain why that is the case.

Since ψ is a linear transformation, we have

$$\int_0^1 f(x) dP(x) = \psi \int_0^1 z(x) dP(x).$$

To calculate this integral, we will proceed component-wise (recall that z is a vector). Thus, we need to calculate, for every $i \in \{1, \ldots d\}$

$$\int_0^1 \frac{e^{c_i}(c_i - 1) + 1}{c_i(e^{c_i} - 1)} dP(x)$$

where $c_i = \frac{1}{\sqrt{d}} \langle (\psi_2)_i, \psi_1 x \rangle$.

This is where the computation gets a bit tricky. The integral calculators that we tried cannot produce an explicit formula for this integral in terms of ψ_1, ψ_2, and we need to solve it by hand. Here is one approach.

Observe that $c_i = w_1^i x_1 + w_2^i x_2 + \cdots + w_d^i x_d$, where w_j^i are constants that depend only on ψ_1 and ψ_2. Thus the integral can be rewritten as:

$$\int_0^1 \int_0^1 \cdots \int_0^1 \frac{e^{w_1^i x_1 + \cdots + w_d^i x_d}(w_1^i x_1 + \ldots w_d^i x_d - 1) + 1}{(w_1^i x_1 + \ldots w_d^i x_d)(e^{w_1^i x_1 + \cdots + w_d^i x_d w_d^i x_d} - 1)} dx_1 \ldots dx_d$$

Let us proceed by the change of variables method. Let

$$\begin{aligned}
u_1 &= w_1^i x_1 + w_2^i x_2 + \cdots + w_d^i x_d \\
u_2 &= x_2 \\
u_3 &= x_3 \\
&\cdots \\
u_d &= x_d
\end{aligned}$$

which has the determinant $\frac{1}{w_1^i}$. Since $x_j \in [0, 1]$ for all j, the new limits are:

$$\begin{aligned}
w_2^i u_2 + \cdots + w_d^i u_d \leq& u_1 \leq w_1^i + w_2^i u_2 + \cdots + w_d^i u_d \\
0 \leq& u_2 \leq 1 \\
&\cdots \\
0 \leq& u_d \leq 1
\end{aligned}$$

and the integral becomes:

$$\left| \frac{1}{w_1^i} \right| \int_0^1 \int_0^1 \cdots \int_{w_2^i u_2 + \cdots + w_d^i u_d}^{w_1^i + w_2^i u_2 + \cdots + w_d^i u_d} \frac{e^{u_1}(u_1 - 1) + 1}{u_1(e^{u_1} - 1)} du_1 \ldots du_d$$

By Fubini's theorem, we can evaluate the integrals iteratively. To compute the inner most integral, we first find the antiderivative. We substitute $s = e^{u_1}$, with $ds/dsu_1 = s$ so that we have

$$\begin{aligned}
\int \frac{e^{u_1}(u_1-1)+1}{u_1(e^{u_1}-1)} du_1 &= \int \frac{s(\log(s)-1)+1}{\log(s)(s-1)} \frac{1}{s} ds \\
&= \int \frac{\log(s)-1}{\log(s)(s-1)} + \frac{1}{\log(s)(s-1)s} ds \\
&= \int \frac{1}{s-1} - \frac{1}{\log(s)(s-1)} + \frac{1}{\log(s)(s-1)s} ds \\
&= \int \frac{1}{s-1} - \frac{1}{s\log(s)} ds \\
&= \log(|s-1|) - \log(|\log(s)|) \\
&= \log(|e^{u_1}-1|) - \log(|u_1|)
\end{aligned}$$

plugging this back into the original integral, we have

$$\left|\frac{1}{w_1^i}\right| \int_0^1 \dots \int_0^1 \left[\log(|e^{u_1}-1|) - \log(|u_1|)\right]_{w_2^i u_2 + \dots + w_d^i u_d}^{w_1^i + w_2^i u_2 + \dots + w_d^i u_d} du_2 \dots du_d$$

The integral now looks much simpler than the original one. Let us first consider the case when $d = 2$. Using linearity, we can split the integral into:

$$\begin{aligned}
I_1 &= \left|\frac{1}{w_1^i}\right| \left(\int_0^1 \log(|e^{w_1^i + w_2^i u_2} - 1|) du_2 - \int_0^1 \log(|e^{w_2^i u_2} - 1|) du_2 \right) \\
I_2 &= \left|\frac{1}{w_1^i}\right| \left(\int_0^1 \log(|w_1^i + w_2^i u_2|) du_2 - \int_0^1 \log(|w_2^i u_2|) du_2 \right)
\end{aligned}$$

The computation of I_2 is easy, and we have:

$$\begin{aligned}
&\int_0^1 \log(|w_2^i u_2|) du_2 = \log(|w_2^i|) - 1 \\
&\int_0^1 \log(|w_1^i + w_2^i u_2|) du_2 = \frac{1}{w_2^i} \left((w_1^i + w_2^i) \log(|w_1^i + w_2^i|) - w_1^i \log(|w_1^i|) - w_2^i \right)
\end{aligned}$$

However, the computation of I_1 is not so easy. Observe that if we perform the substitution $s = e^{w_1^i + w_2^i u_2}$ with $du/ds = w_2^i s$, the integral becomes

$$\int_0^1 \log(|e^{w_1^i + w_2^i u_2} - 1|) du_2 = \frac{1}{w_2^i} \int_{e^{w_1^i}}^{e^{w_1^i + w_2 i}} \frac{\log(|s-1|)}{s} ds$$

which is a very famous integral. The Spence's function, or, equivalently, the dilogarithm, is defined $\forall z \in \mathbb{C}$ as

$$Li_2(z) = -\int_0^z log(1-z)/zdz$$

and it is the analytic continuation for $|z|> 1$ of the series expansion of $-log(1-z)/z$, given by

$$-log(1-z)/z = \left(1 + \frac{s}{2} + \frac{s^2}{3} + \frac{s^3}{4} + \dots\right)$$

because if we integrate term by term, the integral from 0 to z is

$$-\left(z + \frac{z^2}{4} + \frac{z^3}{9} + \frac{z^4}{16} + \dots\right) = Li_2(z)$$

Now, omitting some steps, we arrive at the final formula

$$\begin{aligned}&\Big(\int_0^1 zdP(x)\Big)_i = \\ &Re\Big(\frac{1}{w_1^i}\Big(\frac{1}{w_2^i}\Big(-Li_2(e^{w_1^i+w_2^i}) + Li_2(e^{w_1^i}) + Li_2(e^{w_2}) - Li_2(1)\Big) \\ &+ \log(w_2^i) - \log(w_1^i + w_2^i)(\frac{w_1^i}{w_2^i} + 1) + \frac{w_1^i}{w_2^i}\log(w_1^i)\Big)\Big)\end{aligned}$$

where *Re* denotes the real part of a complex number. This formula can be easily verified numerically using available integral calculators.

Although this may seem like a closed form solution, actually it is not. The values of the dilogarithm are *not known analytically* except for some particular values, like $Li_2(1) = \pi^2/6$, and in most cases, the function is computed using numerical methods. Thus all we have managed to do in this long computation is express the original integral in terms of another integral, whose value cannot be computed analytically. This suggests that in fact, there does not exist a closed form solution to the integral we are trying to compute. This illustrates a crucial point. While the theory, in particular Lemma 6.7, gives us an exact limit to which our models converge, this limit *cannot* be obtained analytically, and we *must* rely on approximate, numerical techniques.

Nevertheless, we can verify the above formula experimentally. Figure 6.1 shows the supremum distance between the the computed formula and the actual output of

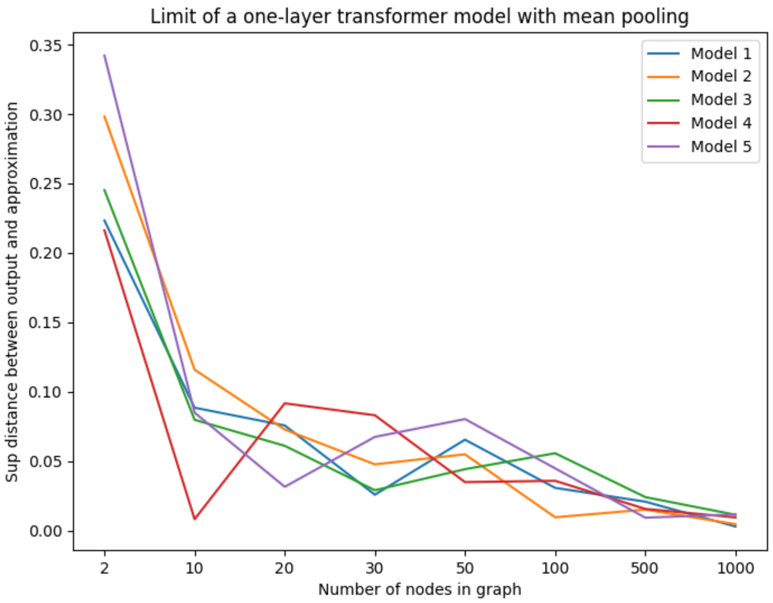


Figure 6.1: Comparison of the output of one-layer transformers with the computed theoretical limit. The plots show convergence to the predicted limit.

a transformer with mean pooling, for 5 different models, as the graph size increases, and the node features are sampled from the uniform distribution on $[0,1] \times [0,1]$. The output dimension is 8, but this was chosen arbitrarily. The dilogarithm function was computed using the Python *mpmath* library. We can see that in all cases, the difference converges to 0, which confirms our computation.

Note that the computed formula was only in the case when $d = 2$. The situation becomes even more complex if the input dimension d is larger than 2, as is usually the case. One approach to approximate the limit would be to perform a similar computation as in the $d = 2$ case, and obtain a formula which would now be in terms of the *polylogarithm function*, and evaluate this numerically. This is possible, but obtaining a recursive formula in terms of d is quite difficult, and not feasible for large d. Another approach is to use a Python package which can solve integrals, like scipy. However, in our experiments, the computation was extremely slow for input dimension larger than 4, which made this technique completely infeasible. We could also try estimating the integral using Monte Carlo methods. This turns

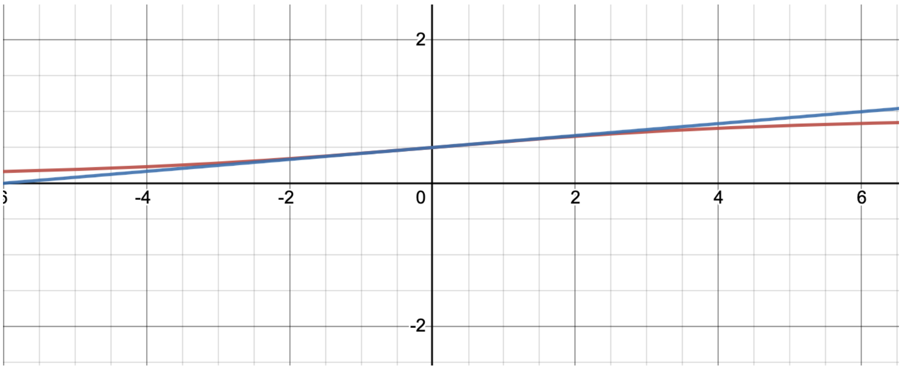


Figure 6.2: Plot of g(t) and 1/2 + t/12.

out to be a much better approach that works even for high dimensional input data, but it will run into computational issues once we start considering more layers.

The solution is to use a Taylor approximation of the function $g(t) = \frac{e^t(t-1)+1}{t(e^t-1)}$. Figure 6.2 shows the plot of $g(t)$ in red and of $1/2 + t/12$ in blue. We can see that $g(t)$ looks like a linear function around the point $t = 0$. Indeed, if we consider the Taylor approximation at the point $t = 0$, we have:

$$\frac{e^t(t-1)+1}{t(e^t-1)} = 1/2 + t/12 - t^3/720 + t^5/30240 + O(t^6)$$

and we can notice that the coefficients for the terms t^3, t^5 and higher are very small. This suggests that we might actually obtain results close to the true value, if we replace $g(t)$ by its first-order approximation

$$\frac{e^t(t-1)+1}{t(e^t-1)} \approx 1/2 + t/12$$

Lemma 6.12

Let $\psi_1, \psi_2, \psi \in \mathbb{R}^{d_1 \times d}$ be the weight matrices of a one-layer transformer. Suppose that the initial node features are sampled independently from $U([0,1]^d)$. Then the first order approximation of the continuous counterpart (Definition 6.3) is given by

$$f(x) \approx \psi(\frac{1}{2} + \frac{c}{12})$$

where $c = \frac{1}{\sqrt{d}}\psi_2^T\psi_1 x$

Proof. Follow directly from Thereom 6.9 and the Taylor approximation of $\frac{e^t(t-1)+1}{t(e^t-1)}$. □

This allows us to compute its expectation with the approximation instead. This is now very simple. We have

Lemma 6.13

Let $\psi_1, \psi_2, \psi \in \mathbb{R}^{d_1 \times d}$ be the weight matrices of a one-layer transformer. Suppose that the initial node features are sampled independently from $U([0,1]^d)$. Then the first order approximation of the expectation of the continuous counterpart (Definition 6.3) is given by

$$\mathbb{E}[f(X)] \approx \frac{1}{2}\psi\mathbf{1} + \frac{1}{24\sqrt{d}}\psi\psi_2^T\psi_1\mathbf{1}$$

Proof. If we let $c = \frac{1}{\sqrt{d}}\psi_2^T\psi_1 X$ and $z_i = \frac{e^{c_i}(c_i-1)+1}{c_i(e^{c_i}-1)}$, by Theorem 6.9 we have $f(X) = \psi z$. Thus by Lemma 2.17, we have $\mathbb{E}[f(X)] = \mathbb{E}[\psi z] = \psi\mathbb{E}[z]$. Now considering a component of $\mathbb{E}[z]$, we have

$$\begin{aligned}
\mathbb{E}[z]_i &= \int_{[0,1]^d} \frac{e^{c_i}(c_i-1)+1}{c_i(e^{c_i}-1)} dP(x) \\
&\approx \int_{[0,1]^d} \frac{1}{2} + \frac{c_i}{12\sqrt{d}} dx \\
&= \int_{[0,1]^d} \frac{1}{2} + \frac{(\psi_2^T\psi_1 x)_i}{12} dx \\
&= \frac{1}{2} + \frac{1}{24\sqrt{d}}(\psi_2^T\psi_1)_i
\end{aligned}$$

which yields the result. □

We can verify this empirically. Figure 6.3 shows the supremum difference between the output of a one-layer transformer and the approximate theoretical limit, computed using the Taylor approximation. Once again, we see the difference converges to 0. Now, the input dimension was chosen to be 8, not 2, as was in the previous case. The number 8 was chosen to achieve a convergence for relatively small graphs. For larger d, the convergence occured for larger graphs.

To summarize, this section, contains several interesting findings. First, we illustrated the difficulty in computing the continuous counterpart in the one layer case, and proved that it is *impossible* to obtain a closed form solution even for a simple distribution, because of the underlying connection to dilogarithmic/polylogarithmic

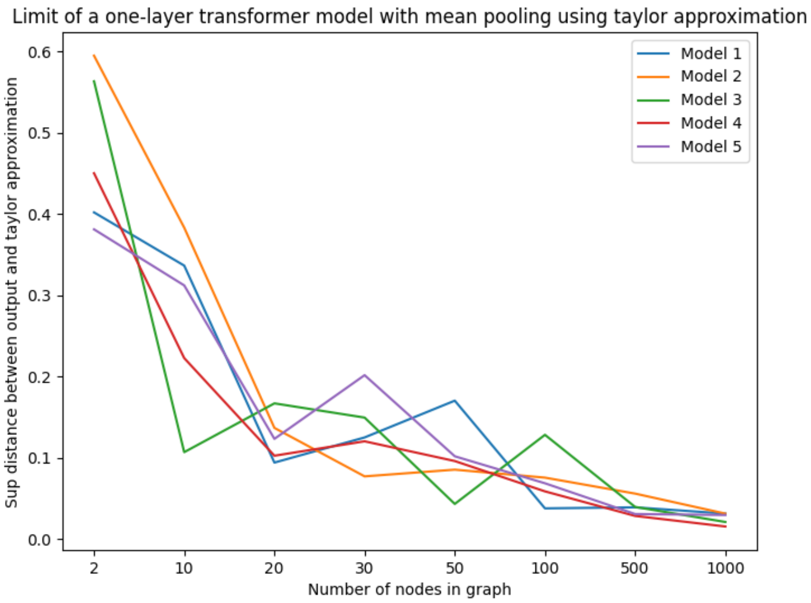


Figure 6.3: Comparison of the output of a one-layer transformer with input dimension $d = 8$ with the theoretical limit computed using the Taylor approximation.

functions, whose values are not known analytically. Then, we developed a numerical technique to work around this fact, and observed that the experiments support the theoretical findings and especially Lemma 6.7.

However, note that we are still considering only one layer. As we will see shortly, the addition of more layers makes the computation even more challenging, and the numerical technique which was presented here must be further improved.

6.5 Theoretical analysis of multiple linear layers

Now that we have a good understanding of what the continuous counterpart is, let us return to theory. Recall that so far, we have only proved the result in the first layer and the purpose of this section is to extend the proof to higher layers. This entire section is very technical. For this reason, let us decided to begin with a slightly less rigorous approach. Otherwise, it would be very difficult to understand why some results are stated the way they are.

Let us understand what needs to be done. At first glance, it looks as though we can use an identical approach as in the one-layer case. We are trying to bound

$\|(S_X)_i^{l+1} - f^{l+1}(X_i)\|_\infty$. Once again, we can use the triangle inequality to get:

$$\begin{aligned}\|(S_X)_i^{l+1} - f^{l+1}(X_i)\|_\infty &\leq \|(S_X)_i^{l+1} - F_n^{l+1}(f^l)(X_i)\|_\infty + \|F_n^{l+1}(f^l)(X_i) - f^{l+1}(X_i)\|_\infty \\ &= \|(S_X)_i^{l+1} - F_n^{l+1}(f^l)(X_i)\|_\infty + a_n^l\end{aligned}$$

where $a_n^l \to 0$ for any fixed $X_i = x_i$ by Theorem 6.4. However, now the situation is more complicated, because we cannot simply apply McDiarmid's inequality to the first term. The reason is that by definition, $(S_X)^{l+1}$ is the output of the F^{l+1}st layer with features $(S_X)^l$, while $F_n^{l+1}(f^l)(X_i)$ is the output of the layer if the features are $f^l(X_i), f^l(X_j)$. Note that this is different to the first layer, because there the features were directly the random variables X_j. For this reason, we must add and subtract $F^{l+1}(f^l(X_i), \{\{f^l(X_j)\}\}_{v_j \in N(v_i)})$ inside the first term, and use the triangle inequality, to get:

$$\begin{aligned}&\leq \left\| (S_X)_i^{l+1} - \sum_{j \in N(v_i)}^{n} \frac{c^{l+1}(f^l(X_i), f^l(X_j))}{\sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(X_i), f^l(X_k))} \psi^{l+1}(f^l(X_j)) \right\|_\infty \\ &+ \left\| \sum_{j \in N(v_i)}^{n} \frac{c^{l+1}(f^l(X_i), f^l(X_j))}{\sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(X_i), f^l(X_k))} \psi^{l+1}(f^l(X_j)) - F_n^{l+1}(f^l)(X_i) \right\|_\infty + a_n^l(X_i)\end{aligned}$$

The idea is that now we can use McDiarmid's inequality for the second term exactly as we did in the case of the first layer.

The question remains what we should do about the first term. Here, we will need to use induction. The idea is that if we assume that $(S_X)_i^l$ is 'close' to $f^l(X_i)$ for all i (this would be the induction assumption), then we can deduce that $(S_X)_i^{l+1}$ is 'close' to $F^{l+1}(f^l(X_i), \{\{f^l(X_j)\}\}_{v_j \in N(v_i)})$.

However, this is not as straightforward. Let us understand why. If we introduce the notation:

$$\begin{aligned} y_i &= (S_X)^l_i \\ y'_i &= f^l(X_i) \\ c_j &= c^{l+1}(y_i, y_j) \\ c'_j &= c^{l+1}(y'_i, y'_j) \\ c &= \sum_{k \in N(v_i)} c_k \\ c' &= \sum_{k \in N(v_i)} c'_k \end{aligned}$$

Then if we follow exactly the same steps as in the proof of Theorem 6.5 (See appendix A), we have that

$$\left\| (S_X)^{l+1}_i - \sum_{j \in N(v_i)}^{n} \frac{c^{l+1}(f^l(X_i), f^l(X_j))}{\sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(X_i), f^l(X_k))} \psi^{l+1}(f^l(X_j)) \right\|_\infty$$
$$\leq \frac{\beta}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \beta \|\psi\|_\infty \|y_j - y'_j\|_\infty + \|\psi\|_\infty \|f^l\|_\infty |c_j - c'_j| + \|\psi\|_\infty \|f^l\|_\infty |c_k - c'_k| \tag{†}$$

By the induction assumption, we assume that $\|y_j - y'_j\|_\infty$ is 'small', so the first term in the sum can be handled easily. However, it is not clear how we should handle the terms $|c_j - c'_j|$ and $|c_k - c'_k|$. We would need a result saying that these are also 'small' provided that $\|y_i - y'_i\|_\infty$,$\|y_j - y'_j\|_\infty$ are 'small'. This is exactly the purpose of the following lemma. Crucially, this lemma is what makes the proof work for concrete models.

Lemma 6.14

Let $l \geq 1$ and let $S \subset \mathbb{R}^{d^l}$ be bounded. Then $\exists \lambda^l > 0$ such that $\forall x, y, x', y' \in S$

$$|c^l(x, y) - c^l(x', y')| \leq \lambda^l (\|x - x'\|_\infty + \|y - y'\|_\infty)$$

Proof. (In the Transformer case)

Recall that by definition of c^l, we are trying to show

$$|\exp(\frac{\langle \psi^l_1(x), \psi^l_2(y) \rangle}{\sqrt{d^l}}) - \exp \frac{(\langle \psi^l_1(x'), \psi^l_2(y') \rangle}{\sqrt{d^l}})| \leq \lambda^l (\|x - x'\|_\infty + \|y - y'\|_\infty)$$

Define $s : S \times S \to \mathbb{R}$ by $s(x,y) = \frac{\langle \psi_1^l(x), \psi_2^l(y)\rangle}{\sqrt{d^l}}$. Since S is bounded, $I = s(S \times S)$ is also bounded, which follows from the Cauchy-Schwarz inequality and boundedness of the linear operators. By the Mean Value theorem, we have:

$$|\exp(s(x,y)) - \exp(s(x',y'))| \leq \max_{t \in I} |\exp(t)| |s(x,y) - s(x',y')|$$

using the fact that $\frac{d}{dt}e^t = e^t$. Since I is bounded we have $\max_{t\in I}|\exp(t)| < \infty$. Let us set $M = \max_{t\in I}|\exp(t)|$, and let's focus on the term $|s(x,y) - s(x',y')|$. If we add and subtract $s(x,y')$ and apply the triangle inequality, we obtain

$$\begin{aligned}|\exp(s(x,y)) - \exp(s(x',y'))| \leq M \Bigg(&\left| \frac{\langle \psi_1^l(x), \psi_2^l(y) - \psi_2^l(y')\rangle}{\sqrt{d^l}} \right| + \\ &\left| \frac{\langle \psi_2^l(y'), \psi_1^l(x) - \psi_1^l(x')\rangle}{\sqrt{d^l}} \right| \Bigg)\end{aligned}$$

Using linearity and the Cauchy-Schwarz inequality, this is

$$\begin{aligned}&\leq \frac{M}{\sqrt{d^l}} (\|\psi_1^l(x)\|_2 \|\psi_2^l(y - y')\|_2 + \|\psi_2^l(y)\|_2 \|\psi_1^l(x - x')\|_2) \\ &\leq \frac{M}{\sqrt{d^l}} (\|\psi_1^l\|_2 \|x\|_2 \| \|\psi_2^l\|_2 \| \|y - y'\|_2 + \|\psi_1^l\|_2 \|y\|_2 \| \|\psi_2^l\|_2 \| \|x - x'\|_2)\end{aligned}$$

But since S is assumed to be bounded, $\|x\|_2$ and $\|y\|_2$ are bounded. Thus we get

$$\begin{aligned}&\leq C_1 \|y - y'\|_2 + C_2 \|x - x'\|_2 \\ &\leq \lambda^l (\|x - x'\|_\infty + \|y - y'\|_\infty)\end{aligned}$$

by setting λ^l to be the maximum of C_1, C_2. □

This lemma, and its proof, are what makes the result work for the transformer architecture. In the paper [18], a result like Lemma 6.14 was never proved. Instead, the authors *assume* a result like Lemma 6.14, but they do not discuss the validity of this assumption.

The following lemma is a slight modifications of a Lemma in [18]. In particular, the Lemma is stated and proved using Lemma 6.14, which was not done in [18].

Lemma 6.15 ([18]).
Let $l \geq 1$ and suppose $(S_X)_i^l$ and $f^l(X_i)$ take values in a bounded set S, forall i.

Suppose the attention mechanism satisfies the claim in Lemma 6.14. Then there exist constants C_1^l and C_2^l such that

$$\left\| (S_X)_i^{l+1} - \sum_{j \in N(v_i)}^{n} \frac{c^{l+1}(f^l(X_i), f^l(X_j))}{\sum_{k \in N(v_i)}^{n} c^{l+1}(f^l(X_i), f^l(X_k))} \psi^{l+1}(f^l(X_j)) \right\|_\infty$$
$$\leq C_1^l \| (S_X)_i^l - f^l(X_i) \|_\infty + C_2^l \max_{m \neq i} \| (S_X)_m^l - f^l(X_m) \|_\infty$$

Proof. [18] Apply Lemma 6.14 to †. □

Note that the requirement for S to be bounded comes from Lemma 6.14. For the moment, ignore this. We will discuss the validity of this assumption later.

Theorem 6.16 ([18]).
Let $L \geq 1$. Let C_1^l, C_2^l be the constants as described in Lemma 6.15. Let $A_l^L = \prod_{k=l+1}^{L} (C_1^k + C_2^k)$ for $l \leq L$. Let

$$H^L(\rho) = \sum_{l=1}^{L} A_l^L \left[D_n^l \sqrt{\frac{n-1}{2} \ln \frac{2^{L+2-l} n d_l}{\rho}} + a_n^l \right]$$

where d_l represents the output dimension of the l-th layer, and D_n^l are as in Theorem 6.5.

Then, with probability at least $1 - \rho$, we have

$$\max_i \| (S_X)_i^L - f^L(X_i) \|_\infty \leq H^L(\rho)$$

Proof. [18] The proof of this theorem uses the ideas that we informally described before. The calculation is quite tedious and is omitted. □

The proof of Theorem 6.16 relies on Lemma 6.15, where we assume that $(S_X)_i^l$ and $f^l(X_i)$ take values in a bounded set S. The question is whether this assumption is feasible. Recall that by Lemma 6.1, f^l is a bounded function, meaning it takes values in a bounded set, which is what we need. However, the case of $(S_X)_i^l$ is not as immediate. The difference between the two is that by definition, f^l does not depend on the graph size, whereas $(S_X)_i$ does, and we need it to be bounded even as n increases. Luckily, we can prove the claim quite easily by induction. However, notice that once again, we will rely on the fact that the features are sampled from a compact space. As you have probably noticed by now, this compactness assumption is absolutely crucial in almost everything we have done so far.

Lemma 6.17

Let $L \geq 1$ and fix an arbitrary node i. Then for any $n \in \mathbb{N}$ where n is the number of nodes in the fully connected graph, $(S_X)_i^L$ is bounded.

Proof. By induction.

For the induction base case, we consider the first layer. We have

$$\begin{aligned}
\|(S_X)_i^1\|_\infty &= \|\sum_{j=1}^{n} \frac{c(X_i, X_j)}{\sum_{k=1}^{n} c(X_i, X_k)} \psi^1(X_j)\|_\infty \\
&\leq \sum_{j=1}^{n} \|\frac{c(X_i, X_j)}{\sum_{k=1}^{n} c(X_i, X_k)} \psi^1(X_j)\|_\infty = \sum_{j=1}^{n} \frac{c(X_i, X_j)}{\sum_{k=1}^{n} c(X_i, X_k)} \|\psi^1(X_j)\|_\infty \\
&\leq \|\psi^1\|_\infty M^1 \sum_{j=1}^{n} \frac{c(X_i, X_j)}{\sum_{k=1}^{n} c(X_i, X_k)} = \|\psi^1\|_\infty M^1 < \infty
\end{aligned}$$

where M^1 is an upper bound for $x \in X$. This upper bound exists precisely because we assume that X is a compact (closed and bounded) set. This concludes the base case. The induction step is almost identical to the above. Suppose that $\|(S_X)_j^l\|_\infty < M^l$ for some constant M^l. Then for $l+1$, we have

$$\begin{aligned}
\|(S_X)_i^{l+1}\|_\infty &= \|\sum_{j=1}^{n} \frac{c((S_X)_i^l, (S_X)_j^l)}{\sum_{k=1}^{n} c((S_X)_i^l, (S_X)_k^l)} \psi^l(S_X)_j^l\|_\infty \\
&= \sum_{j=1}^{n} \frac{c((S_X)_i^l, (S_X)_j^l)}{\sum_{k=1}^{n} c((S_X)_i^l, (S_X)_k^l)} \|\psi^l(S_X)_j^l\|_\infty \\
&\leq \|\psi^l\|_\infty M^l \sum_{j=1}^{n} \frac{c((S_X)_i^l, (S_X)_j^l)}{\sum_{k=1}^{n} c((S_X)_i^l, (S_X)_k^l)} = \|\psi^l\|_\infty M^l < \infty
\end{aligned}$$

as required. □

Note that this applies to a general (non-negative) attention mechanism, not only to that of a transformer.

Corollary 6.3

Let $\mu = \mathbb{E}[f^L(X_1)]$. Then $\frac{1}{n}\sum_{i=1}^{n}(S_X)_i^L$ converges to μ in probability.

Proof. Same as Corollary 6.7 but now we apply Lemma 6.14, Lemma 6.15 and Theorem 6.6. □

And finally, we obtain the zero-one law

Corollary 6.4

Let $X \subset \mathbb{R}^d$ be compact, let P be a probability measure on X, and let (X_n) be a sequence of independent random variables with distribution given by P. Then a transformer with L layers, a mean pooling layer, and a non-splitting classifier C, satisfies the zero-one law with respect to (X_n) as the initial node features, on fully-connected graphs.

Proof. Follows from Corollary 6.3. □

This concludes the theoretical results of this section. We presented an approach inspired by the paper [18], extended the results to the concrete instance of the transformer architecture and obtained a novel zero-one law.

6.6 Experiments with multiple layers

We will now verify Corollary 6.3 experimentally. Once again, we need to compute the continuous counterpart, and as was shown previously, we must do this numerically. Let us recall that the continuous counterpart is defined as

$$f^{l+1}(x) = \int_{y \in X} \frac{c^l(f^l(x), f^l(y))}{\int_{t \in X} c^l(f^l(x), f^l(t)) dP(t)} \psi^l(f^l(y)) dP(y)$$

In the previous section, we managed to show that

$$f^1(x) = \psi \left(\frac{e^c(c-1)+1}{c(e^c-1)} \right)$$

where $c = \frac{1}{\sqrt{d}} \psi_2^T \psi_1 x$. It is clear that finding f^l for $l \geq 2$ analytically is not feasible, because the integral is hard to compute, and we must rely on numerical techniques.

In the one-layer case, we mentioned that solving the integrals using scipy is not a good idea, and we mentioned that even the Monte Carlo approach fails for higher layers. The reason is that because f^l are defined recursively, in order to estimate f^{l+1}, we need to estimate f^l, and in order to do that, we need to estimate f^{l-1}, and so on all the way to f^1, for which we have an analytical solution. If we choose to estimate each integral using only 10 samples, the total number of iterations is 2×10^l, which blows up very quickly. Clearly this method is not feasible for computing the continuous counterpart in higher layers.

Once again, we will rely on the Taylor approximation, but now, the computation is slightly more challenging. This is described in the following lemma.

Lemma 6.18

Let $L \geq 1$ and for every l where $1 \leq l \leq L$, let $\psi_1^l, \psi_2^l, \psi^l \in \mathbb{R}^{d_l \times d_{l-1}}$ be the weight matrices in the l-th layer of a transformer. Let f^l denote the l-th continuous counterpart (Definition 6.3). If $f^l(x) \approx a^l + A^l x$ for some vector a^l and some matrix A^l, then the first order approximation of f^{l+1} is of the form $f^{l+1}(x) \approx a^{l+1} + A^{l+1}x$ where

$$
\begin{aligned}
a^{l+1} &= \psi^{l+1}\left(a^l + \frac{1}{2}A^l\mathbf{1} + \frac{1}{12\sqrt{d_l}}A^l(\psi_2^{l+1}A^l)^T\psi_1^{l+1}a^l\right) \\
A^{l+1} &= \psi^{l+1}A^l\left(\frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}A^l\right)
\end{aligned}
$$

where $\mathbf{1}$ denotes a vector of ones.

Before we prove the lemma, let us discuss its consequences. In the one layer case, we proved that $f^1(x) \approx \psi(\frac{1}{2} + \frac{1}{12\sqrt{d}}\psi_2^T\psi_1 x)$ (Theorem 6.12). Thus, in view of the Lemma 6.18, we can set $A^1 = \frac{1}{2}\psi\mathbf{1}$ and $a^1 = \frac{1}{12}\psi_2^T\psi_1 x$ and we can repeatedly apply the lemma through the layers, to obtain a first order approximation for any f^l. Additionally, implementing this computation is very straightforward. At each layer, we compute A^l, a^l using the formula presented in the lemma, which is just a simple product of matrices, and pass it to the next iteration.

However, at this stage, we don't know whether this approximation will be any good - it could be the case that as we proceed to higher layers, the approximation gets worse and worse. Luckily, we will see that this is not the case, and the approximation is very accurate. Let us now prove the lemma.

Proof Of Lemma 6.18. By definition of f^{l+1}, we have

$$
f^{l+1}(x) = \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}f^l(x), \psi_2^{l+1}f^l(y)\rangle)\psi^{l+1}f^l(y)dy}{\int \exp((\frac{1}{\sqrt{d_l}}\langle(\psi_1^{l+1}f^l(x), \psi_2^{l+1}f^l(t)\rangle)dt}
$$

Plugging in $f^l(x) = \psi^l\left(a^l + A^l x\right)$ this is

$$f^{l+1}(x) = \frac{\int \exp((\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(a^l + A^l y\right)\rangle\psi^{l+1}\left(a^l + A^l y\right)dy}{\int \exp((\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(a^l + A^l t\right)\rangle dt}$$

Using linearity of the inner product and properties of the exponential function, we can cancel out the constant $\exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}a^l\rangle)$ and we have

$$f^{l+1}(x) = \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(A^l y\right)\rangle\psi^{l+1}\left(a^l + A^l y\right)dy}{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}A^l t\rangle dt}$$

and we can further split this into two terms:

$$\begin{aligned}
I_1(x) &= \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(A^y\right)\rangle\psi^{l+1}a^l dy}{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}A^l t\rangle dt} \\
I_2(x) &= \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(A^l y\right)\rangle\psi^{l+1}A^l y dy}{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}A^l t\rangle dt}
\end{aligned}$$

Clearly I_1 is simply $\psi^{l+1}a^l$. For I_2 it remains to notice that we have already computed this type of integrals in Lemma 6.11. Rewriting, this is

$$f^{l+1}(x) = \psi^{l+1}A^l\left(\frac{\int \exp(c_1 y_1 + c_2 y_2 + \dots c_d y_d) y dy}{\int \exp(c_1 y_1 + c_2 y_2 + \dots c_d y_d))}\right)$$

where $c(x) = \frac{1}{\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}(a^l + A^l x)$. By Lemma 6.10 and Lemma 6.11, we have

$$I_2(x) = \psi^{l+1}A^l\left(\frac{e^c(c-1)+1}{c(e^c-1)}\right)$$

Using the Taylor series we have:

$$\begin{aligned}
I_2(x) &\approx \psi^{l+1}A^l(\frac{1}{2}\mathbf{1} + \frac{1}{12}c(x)) \\
&= \psi^{l+1}A^l(\frac{1}{2}\mathbf{1} + \frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}(a^l + A^l x)) \\
&= \psi^{l+1}A^l\left(\frac{1}{2}\mathbf{1} + \frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}a^l\right) \\
&+ \psi^{l+1}A^l\left(\frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}A^l x\right)
\end{aligned}$$

now writing $f^{l+1}(x) = I_1 + I_2$, we have:

$$\begin{aligned} f^{l+1}(x) &\approx \psi^{l+1}\left(a^l + \frac{1}{2}A^l\mathbf{1} + \frac{1}{12\sqrt{d_l}}A^l(\psi_2^{l+1}A^l)^T\psi_1^{l+1}a^l\right) \\ &+ \psi^{l+1}A^l\left(\frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}A^l x\right) \\ &= a^{l+1} + A^{l+1}x \end{aligned}$$

as required. □

Finally, observe that using the lemma, we can approximately compute $\mathbb{E}[f^{l+1}(x)]$ as

$$\mathbb{E}[f^{l+1}(x)] \approx a^{l+1} + \frac{1}{2}A^{l+1}(\mathbf{1}) \approx f^{l+1}(\frac{1}{2}\mathbf{1})$$

where the $1/2$ comes from the integral of x on $[0,1]^d$.

Let us now examine the accuracy of these predictions. To start, let us take 5 transformer models with, say, *8 layers* and a final mean pooling layer, and examine whether their output is close to $f^8(\frac{1}{2}\mathbf{1})$, which is computed using the Taylor approximation. As before, we set the input dimension, hidden dimension output dimension to 8, but note that this is arbitrary.

Figure 6.4 shows that indeed the difference between the output and the continuous counterpart goes to zero. Let us also show the actual outputs. For example, for model number 5, the predicted value, and the actual output are

$$\text{output} = \begin{bmatrix} 28.17432864 \\ -13.62785145 \\ 3.73143464 \\ 1.42475506 \\ -15.92777957 \\ 23.48883893 \\ -9.96977534 \\ 11.37577893 \end{bmatrix} \quad f^8(1/2) = \begin{bmatrix} 27.96426481 \\ -13.61058432 \\ 3.74093775 \\ 1.37021914 \\ -15.79265324 \\ 23.2886607 \\ -9.87223274 \\ 11.24106646 \end{bmatrix}$$

Note that now, we are considering a graph of size 10000, which is very large, and in fact it takes a couple of seconds to compute the actual output. In contrast, the approximation is computed instantly, and does not even look at the sampled node features. Furthermore, looking at the plot, we see that model number 5 was close already for graphs of size 500, which suggest that some models can converge quickly even if they have a higher number of layers.

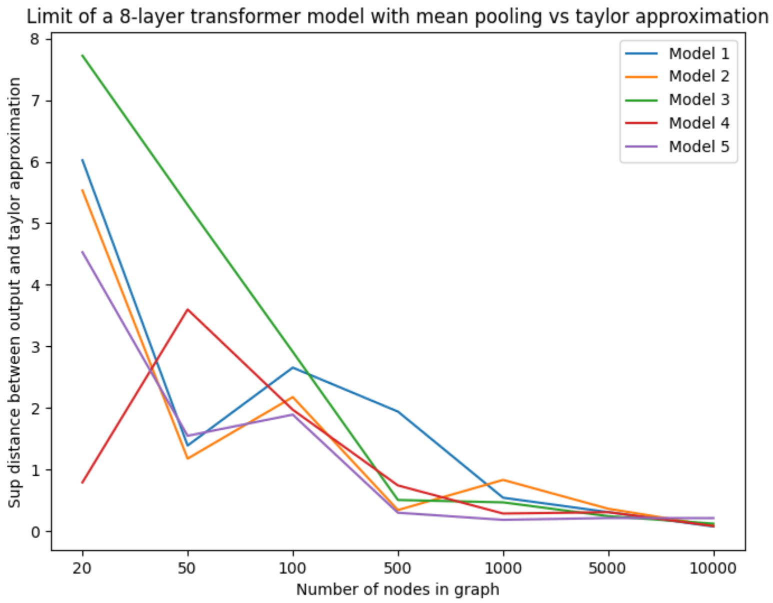


Figure 6.4: Comparison of the output of 5 transformers with 8-layers with the expectation of the 8-th continuous counterpart, computed using the Taylor approximation.

6.6.1 Experiments with other distributions

Now that we have a good understanding of the phenomenon in the Uniform $[0, 1]^d$ case, we can start examining different distributions. After all, forcing the node features to come from the Uniform$[0, 1]^d$ distribution is quite restrictive. However, recall that our theory doesn't support an arbitrary distribution, because the node features must be sampled from a compact space X. For this reason, we will study a general Uniform distribution.

Once again, we must figure out a way to compute the continuous counterpart. Luckily, this will be almost identical to the Uniform $[0, 1]$ case. We only need to tweak the previous results slightly.

Lemma 6.19

Let $d \in \mathbb{R}$, $c_i \neq 0$ and $a_i \leq b_i$ for $i = 1, \ldots d$. Let

$$I_d = \int_{a_d}^{b_d} \ldots \int_{a_1}^{b_1} \exp(c_1 x_1 + c_2 x_2 + \ldots c_d x_d) dx_1 dx_2 \ldots dx_d$$

Then $I_d = \prod_{i=1}^{d} \frac{e^{c_i b_i} - e^{c_i a_i}}{c_i}$

Proof. Identical to 6.10. □

Lemma 6.20

Let $d \in \mathbb{N}$, $c_i \neq 0$ for $i = 1, \dots d$ and $w \neq 0$. Let

$$I_d = \int_{a_d}^{b_d} \dots \int_{a_1}^{b_1} \exp(c_1 x_1 + c_2 x_2 + \dots c_d x_d) w x_1 dx_1 dx_2 \dots dx_d$$

Then $I_d = w \frac{e^{b_1 c_1}(c_1 b_1 - 1) - e^{a_1 c_1}(c_1 a_1 - 1)}{c_1^2} \prod_{j=2}^{d} \frac{e^{c_j b_j} - e^{c_j a_j}}{c_j}$

Proof. Same as 6.11. □

Using these two results, we can calculate the continuous counterpart.

Lemma 6.21

Let $\psi, \psi_1, \psi_2 \in \mathbb{R}^{d_1 \times d}$ be the weight matrices in a one-layer transformer. Let $x \in [a_1, b_1] \times \dots [a_d, b_d]$ be arbitrary and let $c = \frac{1}{\sqrt{d_1}} \psi_2^T \psi_1 x$. Then the continuous counterpart (Definition 6.3) with respect to the $U([a_1, b_1] \times \dots [a_d, b_d])$ distribution, is given by

$$f(x) = \psi z$$

where

$$z_i = \frac{e^{b_i c_i}(c_i b_i - 1) - e^{a_i c_i}(c_i a_i - 1)}{c_i (e^{c_i b_i} - e^{c_i a_i})}$$

Proof. Apply Lemma 6.19 and Lemma 6.20. □

Lemma 6.22

Let $g(t) = \frac{e^{bt}(tb-1) - e^{at}(ta-1)}{t(e^{tb} - e^{ta})}$*. Then the Taylor approximation of $g(t)$ at $t = 0$ is*

$$g(t) = \psi \left(\frac{a+b}{2} + \frac{t(b-a)^2}{12} - \frac{t^3(b-a)^4}{720} + \frac{t^5(b-a)^6}{30240} + O(t^6) \right)$$

Once again, the coefficients for the terms t^3, t are small, which means we can use the first order approximation

Corollary 6.5

Under the same setup as Lemma 6.21, the first order approximation of the continuous counterpart is given by

$$f^1(x) \approx \psi\left(\frac{a+b}{2} + \frac{c(x) * (b-a)^2}{12}\right)$$

where $*$ denotes element wise product.

Now we can extend the approximation to higher layers, similarly as was done before.

Lemma 6.23

Let $l \geq 1$, $L = (a_1, a_2 \ldots a_d)$, $U = (b_1, b_2 \ldots b_d)$. Suppose that $f^l(x) \approx a^l + A^l x$ for some vector a^l and some matrix A^l. Then the first order approximation of f^{l+1} is of the form $f^{l+1}(x) \approx a^{l+1} + A^{l+1}x$ where

$$\begin{aligned} a^{l+1} &= \psi^{l+1}\left(a^l + \frac{1}{2}A^l(L+U) + \frac{1}{12\sqrt{d_l}}A^l(\psi_2^{l+1}A^l)^T\psi_1^{l+1}a^l\right) \\ A^{l+1} &= \psi^{l+1}A^l\left(\frac{1}{12\sqrt{d_l}}(\psi_2^{l+1}A^l)^T\psi_1^{l+1}((A^l)^T * (U-L)^2)^T\right) \end{aligned}$$

where $*$ denotes element wise product.

Before we prove the lemma a short explanation of the notation is required. The term $((A^l)^T * (U-L)^2)^T$ means that we multiply row i of A by $(U-L)_i^2$. The reason for this notation is that this is how you can easily implement it in Numpy/PyTorch.

Proof. The proof is almost identical to the proof of Lemma 6.18, but differs in the Taylor expansion step. Using the same logic as before, if we let

$$\begin{aligned} I_1(x) &= \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(A^l y\right)\rangle\psi^{l+1}a^l dy}{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}A^l t\rangle)dt} \\ I_2(x) &= \frac{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}\left(A^l y\right)\rangle\psi^{l+1}A^l y dy}{\int \exp(\frac{1}{\sqrt{d_l}}\langle\psi_1^{l+1}\left(a^l + A^l x\right), \psi_2^{l+1}A^l t\rangle)dt} \end{aligned}$$

where the integrals are over the rectangle $[a_1, b_1] \times \ldots \times [a_d, b_d]$, then we know

$$f^{l+1}(x) = I_1(x) + I_2(x)$$

Now $I_1(x)$ is just $\psi^{l+1}a^l$, and for $I_2(x)$, we have

$$I_2(x)_i = \psi^{l+1}A^l \left(\frac{e^{b_i c_i}(c_i b_i - 1) - e^{a_i c_i}(c_i a_i - 1)}{c_i(e^{c_i b_i} - e^{c_i a_i})} \right)$$

where $c(x) = (\frac{1}{\sqrt{d_l}}\psi_2^{l+1}A^l)^T \psi_1^{l+1}(a^l + A^l x)$, by Lemma 6.19 and Lemma 6.20. Now we use the Taylor expansion given in Lemma 6.22, to obtain

$$I_2(x)_i \approx \psi^{l+1}A^l \left(\frac{(a_i + b_i)}{2} + \frac{c_i(a_i - b_i)^2}{12} \right)$$

and the result follows. □

Let us now do some experiments. To begin with, we will set each a_i, b_i so that $b_i - a_i = 1$. Once again let's consider 8 dimensional features in all layers, and let's choose something random for the input distribution, say

$$a_1, a_2 \ldots a_8 = (-4, 1.5, 2.2, 4, 0, -1, -2.7, -0.5)$$
$$b_1, b_2 \ldots b_8 = (-3, 2.5, 3.2, 5, 1, 0, -1.7, 0.5)$$

We will take 5 transformer models with randomly initialized weights, and no nonlinearities in between the layers. For the prediction, we will evaluate f^8 at the mean of the above distribution, which is $\mu = (-3.5, 2, 2.5, 2, 7, 4.5, 0.5, -0.5, -2.2, 0)$

Figure 6.5 shows that 4 models nicely converge, and 1 is slightly off. The "best" model is model 2 is at a final distance of only 0.1068. The predicted output and the actual output were

$$\text{output} = \begin{bmatrix} -185.88071535 \\ -170.54799564 \\ -180.40062931 \\ 29.25255822 \\ -56.43718566 \\ -146.05627772 \\ -40.88151963 \\ 93.0486853 \end{bmatrix} \quad f^8(\mu) = \begin{bmatrix} -185.91399265 \\ -170.61271252 \\ -180.50743291 \\ 29.35773608 \\ -56.48056111 \\ -146.13218433 \\ -40.83049326 \\ 93.02292706 \end{bmatrix}$$

we can see that the prediction is very accurate.

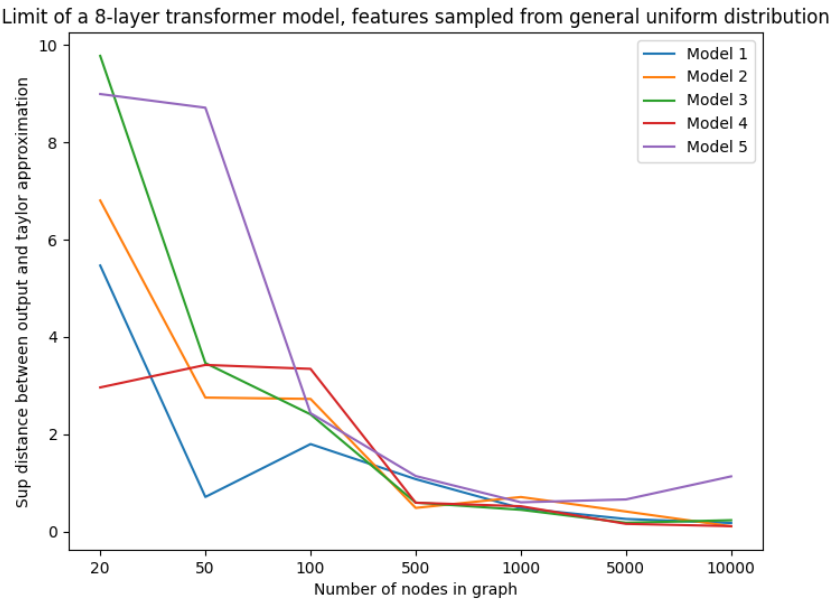


Figure 6.5: Comparison of the output of 5 transformers with 8-layers with the expectation of the 8-th continuous counterpart, computed using the Taylor approximation, with respect to a general uniform distribution.

The worst model is model 5, which as at a distance of 1.1331 from the predicted value. The predicted output and the actual output were

$$\text{output} = \begin{bmatrix} 172.62025343 \\ -202.64470263 \\ 9.14126939 \\ -72.85177321 \\ -41.11725308 \\ 108.5661577 \\ -62.53759354 \\ -126.26521247 \end{bmatrix} \quad f^8(\mu) = \begin{bmatrix} 173.75336338 \\ -203.38658748 \\ 9.19580312 \\ -72.94783811 \\ -41.62508932 \\ 109.44973286 \\ -62.97361769 \\ -126.26092992 \end{bmatrix}$$

The error of 1.1331 comes in the first entry, but note that this is an error of only 0.65%, so the approximation is still quite close to the real output.

We can also consider node features with dimension $d > 8$, and distributions where $b_i - a_i > 1$. We observed that convergence occurs in these cases as well, but is slightly slower, and a larger number of nodes is needed.

To summarize, in this section we developed an experimental approach to numerically compute the continuous counterpart and its mean. We observed that the actual model outputs are very close to this limit even for a high number of layers. This verifies the claim that transformers (as we defined them), satisfy a zero-one law.

6.7 Different attention mechanisms

In the previous sections, we developed a general theoretical framework allowing us to understand the convergence of attention models. We stated, proved, and experimentally verified the zero-one law for the transformer architecture. In this section, we would like to do the same for other attention networks.

Looking back at our theory, we can see that the only places where we actually use the concrete attention mechanism of a transformer, were in the proofs of Lemma 6.1 and Lemma 6.14, where we plugged in $c^l(x,y) = \exp(\frac{\langle \psi_1^l(x), \psi_2^l(y) \rangle}{\sqrt{d^l}})$. This means that if we could prove these Lemmas using a different attention mechanism, the rest of the theory would apply, and we would get the desired zero-one law quite easily. Thus, this section will be dedicated to proving Lemma 6.1 and Lemma 6.14 for different attention mechanisms.

Let us begin with graph attention networks (2.9), where the attention mechanism is defined as $c^l(x,y) = \exp(\text{LeakyReLU}(\langle a^l, \psi^l(x) || \psi^l(y) \rangle)$, where a^l, ψ^l are parameters of the model and $||$ denotes concatenation. We will denote LeakyReLU by LR.

Lemma 6.24

The LeakyReLU with slope $0 < k < 1$ is 1-Lipschitz continuous.

Proof. We need to show

$$|LR(a) - LR(b)| \leq |a - b|$$

We simply need to consider the cases:

$$|LR(a) - LR(b)| = \begin{cases} |a-b|, & \text{if } a, b \geq 0 \\ |a - kb|, & \text{if } a \geq 0, b < 0 \\ |ka - b| & \text{if } a < 0, b \geq 0 \\ k|a-b| & \text{if } a < 0, b < 0 \end{cases}$$

Cases 1 and 4 follow immediately, since $k < 1$. Case 2 follows after noting that $|a - kb|= a - kb < a - b = |a - b|$ since $a > 0, b < 0, k < 1$. The same argument applies for case 3. □

Let us restate Lemma 6.1 and prove it in the GAT case:

Lemma 6.25

If f^l is bounded, then $\exists\alpha^l, \beta^l > 0$ such that $\forall x, y \in X$ we have $\alpha^l \leq c^l(f^l(x), f^l(y)) \leq \beta^l$, and f^{l+1} is bounded.

Proof. (In the case of GATs)
By definition, we have $c^l(f^l(x), f^l(y)) = \exp(\mathrm{LR}(\langle a^l, \psi^l(f^l(x))||\psi^l(f^l(y))\rangle))$. Observe that $|\mathrm{LR(t)}| \leq t$, thus

$$\begin{aligned}|\mathrm{LR}(\langle a^l, \psi^l(f^l(x))||\psi^l(f^l(y))\rangle)| &\leq |\langle a^l, \psi^l(f^l(x))||\psi^l(f^l(y))\rangle)| \\ &\leq \|a^l\|_2\|\psi^l(f^l(x))||\psi^l(f^l(y))\|_2\end{aligned}$$

where we used the Cauchy-Schwarz inequality. Now, by definition, we have that

$$\begin{aligned}\|\psi^l(f^l(x))||\psi^l(f^l(y))\|_2^2 &= \|\psi^l(f^l(x))\|_2^2{+}\|\psi^l(f^l(y))\|_2^2 \\ &\leq (\|\psi^l\|_2\|(f^l(x))\|_2)^2 + (\|\psi^l\|_2\|(f^l(y))\|_2)^2 < \infty\end{aligned}$$

and thus $\|\psi^l(f^l(x))||\psi^l(f^l(y))\|_\infty{<}\ \infty$. Since $\exp(\cdot)$ is strictly positive, there exist α^l, β^l such that $\alpha^l \leq c^l(f^l(x), f^l(y)) \leq \beta^l$, as desired. The conclusion that f^{l+1} is bounded follows the same way as in the transformer case. □

Let us restate Lemma 6.14 and prove it in the GAT case:

Lemma 6.26

Let $l \geq 1$ and let $S \subset \mathbb{R}^{d^l}$ be bounded. Then $\exists\lambda^l > 0$ such that $\forall x, y, x', y' \in S$

$$|c^l(x, y) - c^l(x', y')|{\leq}\ \lambda^l(\|x - x'\|_\infty{+}\|y - y'\|_\infty)$$

Proof. (In the GAT case)

Recall that by definition of c^l, we are trying to show

$$|\exp\left(\mathrm{LR}(\langle a^l, \psi^l(x)||\psi^l(y)\rangle)\right) - \exp\left(\mathrm{LR}(\langle a^l, \psi^l(x')||\psi^l(y')\rangle)\right| \\ \leq \lambda^l(\|x-x'\|_\infty + \|y-y'\|_\infty)$$

where LR denotes a LR with a slope $k < 1$ (the exact value of the slope is not important). Define $s : S \times S \to \mathbb{R}$ by $s(x, y) = (\langle \mathrm{LR}(\langle a^l, \psi^l(x)||\psi^l(y)\rangle\rangle)$. Since S is bounded, $I = s(S \times S)$ is also bounded, which follows from the Cauchy-Schwarz inequality and boundedness of the linear operators. By the Mean Value theorem, we have:

$$|\exp\left(s(x, y)\right) - \exp\left(s(x', y')\right)| \leq \max_{t \in I} |\exp(t)||s(x, y) - s(x', y')|$$

using the fact that $\frac{d}{dt}e^t = e^t$. Since I is bounded we have $\max_{t\in I}|\exp(t)| < \infty$. Let us set $M = \max_{t\in I}|\exp(t)|$, and let's focus on the term $|s(x, y) - s(x', y')|$. If we add and subtract $s(x, y')$, then the triangle inequality gives us

$$|\exp\left(s(x, y)\right) - \exp\left(s(x', y')\right)| \leq M(|s(x, y) - s(x, y')| + |s(x, y') - s(x', y')|)$$

Now, by Lemma 6.24, we can get rid of the LR, to obtain

$$\leq M\left(|\langle a^l, \psi^l(x)||\psi^l(y) - \psi^l(x')||\psi^l(y)\rangle| + |\langle a^l, \psi^l(x)||\psi^l(y') - \psi^l(x)||\psi^l(y')\rangle\right)|$$

where we used the linearity of $\langle\cdot,\cdot\rangle$. Now note that when we concatenate vector A with vector B and vector C with vector B, and compute the difference of these concatenated vectors, it is the same as concatenating $(A - C)$ with $(B - B = 0)$. Thus the above is just:

$$\begin{aligned} &= M\left(|\langle a^l, (\psi^l(x - x'))||0)\rangle| + |\langle a^l, 0||(\psi^l(y - y'))\rangle\rangle|\right) \\ &\leq M\|a^l\|_2\Big(\|\psi^l(x - x')\|_2 + \|\psi^l(y - y')\|_2\Big) \quad \text{Cauchy Schwarz} \\ &\leq M\|a^l\|_2\|\psi^l\|_2(\|x - x'\|_2 + \|y - y'\|_2) \\ &\leq \lambda^l\left(\|x - x'\|_\infty + \|y - y'\|_\infty\right) \end{aligned}$$

where we used norm equivalence (2.15) in the last step. This concludes the proof. □

Finally, we know that the assumption that S is a bounded is valid, because Lemma 6.17 applies to a general attention mechanism.

Let us look at one more common attention mechanism, which it the GATv2 2.10. Here, the proofs of Lemma 6.1 and Lemma 6.14 are very similar as in the case of the standard GAT. Lemma 6.1 once again follows by the Cauchy-Schwarz inequality, and Lemma 6.14 can be proved in exactly the same way, only now we get

$$\begin{aligned}&|\exp\left(s(x,y)\right)-\exp\left(s(x',y')\right)|\leq M(|s(x,y)-s(x,y')|+|s(x,y')-s(x',y')|)\\&=M(|\langle a^l,LR(\psi(x||y))-LR(\psi(x'||y))|+|\langle a^l,LR(\psi(x'||y))-LR(\psi(x'||y'))|\end{aligned}$$

which can be bounded using Lemma 6.24, and the Cauchy-Schwarz inequality, similarly as before. Thus, we can conclude the convergence results also apply to GATv2, and in particular, it satisfies the zero-one law.

Let us now take a step back, and consider the key ingredients of the proofs of the two Lemmas. Notice that in all three cases, we use a very similar argument. In Lemma 6.1, we show that whatever is inside the exponential is bounded, and then we conclude that because the exponential is strictly positive, we can find the two desired constants α^l, β^l, in particular, the lower bound α^l. This suggests that attention mechanisms which use a softmax will satisfy Lemmma 6.1. Thus if we wanted to design an attention mechanism which doesn't satisfy the zero-one law, we shouldn't use softmax. However, this is slightly problematic - the softmax acts as a normalization and is important for numerical stability. This suggests that designing a well-behaved model, which doesn't satisfy the zero-one law is not that straightforward. We will further elaborate on this in due course.

6.8 Nonlinearities

So far, we've only considered versions of attention models which do not have nonlinearities in between the layers. This was done intentionally, in order to make the theory easier to follow. In this section, we will extend the theory to include the nonlinearities. Luckily for us, we already did all the hard work. The theory now only needs a little tweak, and therefore, this section will be very brief. Throughout this section σ will denote a Lipschitz continuous non-linearity with Lipschitz constant k, such as the ReLU, LeakyReLU, tanh, or sigmoid. We will make use of a trivial lemma concerning Lipschitz functions:

Lemma 6.27

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a Lipschitz function with constant k. Let S be a bounded set. Then $\sigma(S)$ is bounded.

Proof. Let $M = \max_{x,y \in S} |x - y|$. Since S is bounded, $M < \infty$. Now fix some $y \in S$, and observe that for any $x \in S$, we have

$$|\sigma(x)| \leq |\sigma(x) - \sigma(y)| + |\sigma(y)| \leq k|x - y| + |\sigma(y)| \leq kM + |\sigma(y)|$$

thus $|\sigma(x)|$ is bounded above by a constant. Since x was arbitrary, $\sigma(S)$ is bounded.

□

We will also need to slightly modify the definition of a continuous counterpart to account for the non-linearities in between the layers.

Definition 6.8

Let $x \in X$. For $l \geq 1$, we recursively define

$$f_\sigma^{l+1}(x) = \sigma\left(\int_{y \in X} \frac{c^l(f_\sigma^l(x), f_\sigma^l(y))}{\int_{t \in X} c^l(f_\sigma^l(x), f_\sigma^l(t)) dP(t)} \psi^l(f_\sigma^l(y)) dP(y) \right)$$

*and $f_\sigma^0(x) = x$. We call f_σ^l the l-th **σ-continuous counterpart**.*

and similarly as in Lemma 6.1, we have the following results

Lemma 6.28

f_σ^l is measurable.

Proof. Same as the proof of 6.3 + composition of measurable functions is measurable.

□

Lemma 6.29

If f_σ^l is bounded, then $\exists \alpha^l, \beta^l > 0$ such that $\forall x, y$ we have $\alpha^l \leq c^l(f_\sigma^l(x), f_\sigma^l(y)) \leq \beta^l$, and f_σ^{l+1} is bounded.

Proof. The first part of the proof is exactly the same as before. For the second part, note that $\forall x \in X$

$$\int_{y\in X} \frac{c^l(f^l_\sigma(x), f^l_\sigma(y))}{\int_{t\in X} c^l(f^l_\sigma(x), f^l_\sigma(t))dP(t)} \psi^l(f^l_\sigma(y))dP(y) \leq \frac{\beta^l \|\psi^l\|_\infty \|f^l_\sigma\|_\infty}{\alpha^l}$$

and now we can apply Lemma 6.27 to conclude that f^{l+1}_σ is bounded. □

Before proceeding formally, let us gain some intuition about the problem. Consider the first layer, and recall that Theorem 6.6 gave us an upper bound for $\|(S_X)_i - f^1(x_i)\|_\infty$ with high probability. We would like to obtain a similar bound once we include nonlinearities into the problem. If we assume that the non-linearity is Lipschitz, we can obtain a bound quite easily. Simply consider what happens if one applies a nonlinearity after the first layer. We have

$$\|\sigma((S_X)_i) - f^1_\sigma(x_i)\|_\infty \leq k\|(S_X)_i - f^1(x_i)\|_\infty$$

where we used the fact that $f^1_\sigma(x_1) = \sigma(f^1(x_1))$. Thus for any C, we have

$$\mathbb{P}(\|\sigma((S_X)_i) - f^1_\sigma(x_i)\|_\infty \leq C) \geq \mathbb{P}(k\|(S_X)_i - f^1(x_i)\|_\infty \leq C)$$

This will allow us to easily obtain an upper bound which holds with high probability. This is exactly why we required the nonlinearity to be Lipschitz.

Let us now state things more formally. In particular, we need to extend Theorem 6.4 to account for σ. For this reason, let us extend Definition 6.5.

Definition 6.9

Let $l \geq 0$. We define $F^{l+1}_{\sigma,n}(f^l_\sigma)(\cdot) : X \to \mathbb{R}^{d_{l+1}}$ by

$$F^{l+1}_{\sigma,n}(f^l_\sigma)(x_i) = \mathbb{E}_{X_j, j\neq i}\left[\sigma(F^{l+1}(f^l_\sigma(x_i), \{\{f^l_\sigma(X_j)\}\}_{v_j\in N(v_i)}))\right]$$

where F^{l+1} is the same as in Definition 6.1

Now we can restate Theorem 6.4

Theorem 6.30

$\|F^{l+1}_{\sigma,n}(f^l_\sigma)(x_i) - f^{l+1}_\sigma(x_i)\|_\infty \to 0$ *as* $n \to \infty$.

Proof. The proof is exactly the same as 6.4, but we use Lemma 6.28 and Lemma 6.29. □

Using these, we can show convergence in the first layer. Note that from now on $(S_X)^l$ will denote represent the pre-activations after l layers. This means that σ is applied in between each of the first $l-1$ layers, so $(S_X)^l = \sigma((S_X)^{l-1})$.

Theorem 6.31

Let $a_n = \|F^1_{\sigma,n}(f^0)(x_i) - f^1_\sigma(x_i)\|_\infty$ and consider $\rho \in (0,1]$. Then

$$\mathbb{P}\left(\|\sigma((S_X)^1_i) - f^1_\sigma(x_i)\|_\infty \leq kD^1_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}} + a_n\right) \geq 1 - \frac{\rho}{n}$$

where k is the Lipschitz constant of σ.

Proof. Fix $X_i = x_1$, We have:

$$\begin{aligned}
&\|\sigma(F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - \sigma(f^1(x_i))\|_\infty \leq \\
&\|\sigma(F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - F_{n,\sigma}(x_i)\|_\infty + \|F_{n,\sigma}(x_i) - \sigma(f^1(x_i))\|_\infty \\
&= \|\sigma(F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - F_{n,\sigma}(x_i)\|_\infty + a_n
\end{aligned}$$

Applying McDiarmid's inequality to $\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - F_n(x_i)\|_\infty$, we have

$$\mathbb{P}\left(\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - F_n(x_i)\|_\infty \leq D^1_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}}\right) \geq 1 - \frac{\rho}{n}$$

and now, by the Lipschitz property:

$$\mathbb{P}\left(\|\sigma(F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})) - F_{n,\sigma}(x_i)\|_\infty \leq kD^1_n\sqrt{\frac{n-1}{2}\ln\frac{2d_1 n}{\rho}}\right) \geq 1 - \frac{\rho}{n}$$

Now we finish off by using the law of total probability, as before. □

Now by Theorem 6.30 and the Theorem 6.5, we obtain convergence in probability. We can extend this to the zero-one law the same way as before.

One can use a similar argument to extend the result to higher layers. However, this is omitted because it is boring, and repetitive. One can also design experiments similarly as before, but unfortunately, the non-linearity makes things quite problematic. Recall that in our experiments, we relied heavily on Taylor expansions. The addition of non-linearities makes a computation like in Lemma 6.23 impossible. One can try approximating the non-linearities, but this introduces inaccuracies. For example, one can use the assumption $ReLU(x+y) \approx ReLU(x) + ReLU(y)$ (which is not true), to make the computation easier, but this results in an approximation which is not very accurate. For this reason, we do not include these experiments.

6.9 A problem with the random graph model

So far in our analysis, we focused only on fully-connected graphs. However, we know that the zero-one law is stated in terms of graphs sampled from the E-R model. This section is dedicated to extending the results to this scenario.

Let us make some important remarks. The paper [18] in fact considers graphs which are not fully-connected. The reason why we discussed only fully-connected graphs in the previous section is that the random graph model that [18] uses is somewhat peculiar.

The paper [18] considers fully-connected graphs where each edge is weighted, using a weight function W. The aggregation scheme and attention then takes this weight into account. The weight can be zero, if that is what one desires, which simulates an edge not being present, thus producing a graph which is, in essence, not fully-connected. The analysis of the convergence is exactly the same, but all definitions and theorems take the weight function into account in the obvious way. So why is this problematic?

The problem lies in the definition of the weight function. In the paper, an edge between nodes i and j is computed as $W(X_i, X_j)$, where X_i, X_j are the node features. This means that the edge weight is a function of the node features, which is not the same as sampling the features and graphs separately. Additionally, it is not at all clear how one would obtain graphs where the edges follow a Bernoulli distribution (so graphs sampled from the Erdős-Rényi model), which is exactly what we are interested in, in view of the zero-one law.

We will now *illustrate* a relatively simple extension of the proof to include graphs sampled from the Erdős-Rényi model and obtain the full zero-one law. Similarly as before, we will begin with the first layer. The proof is a slight modification of the proof of Theorem 6.6 and Corollary 6.1.

Definition 6.10

Let D^1_n, a^1_n be defined as before. Let $\rho > 0$. We define the sequence S^ρ_n by

$$S^\rho_n := \sup_{N \geq n} D^1_N \sqrt{\frac{N-1}{2} \ln(\frac{2d_1 N}{\rho})} + a^1_N$$

Lemma 6.32

For any $\rho > 0$, $\lim_{n\to\infty} S^\rho_n = 0$

Proof. A basic property of sequences is that a sequence (x_n) to converges x iff both $\limsup x_n$ and $\liminf x_n$ converge to x. Since we know $D^1_n\sqrt{\frac{n-1}{2}\ln(\frac{2d_1 n}{\rho})} + a^1_n$ converges to 0, the result follows from this fact. □

The following theorem is a modification of Theorem 6.6, to the case where the neighborhood of the node follows a binomial distribution.

Theorem 6.33

Let $\rho \in (0,1)$ and let $N(v_i)$ denote the neighborhood of node i. Let $n \in \mathbb{N}$ and let $p \in (0,1]$ be such that $np \geq 1$. Suppose $|N(v_i)| \sim Bin(n,p)$, independently from the node features (X_j). Then

$$\mathbb{P}\left(\|(S_X)^1_i - f^1(X_i)\|_\infty \leq S^\rho_{\lfloor \frac{np}{2} \rfloor}\right) \geq (1 - \frac{2\rho}{np})(1 - \exp(-\frac{np}{8}))$$

where $\lfloor \cdot \rfloor$ denotes the floor function.

Proof. Fix $X_i = x_i$. We will denote $N := |N(v_i)|$ and condition on the fact that $N > \frac{np}{2}$ and use the law of total probability. By the same argument as before, we know that if $|N(v_i)| = k$, we have:

$$\mathbb{P}\left(\|F^1(x_i, \{\{X_j\}\}_{v_j \in N(v_i)}) - f^1(x_i)\|_\infty \leq D^1_k \sqrt{\frac{k-1}{2}\ln\frac{2d_1 k}{\rho}} + a_k\right) \geq 1 - \frac{\rho}{k}$$

and thus

$$\mathbb{P}\left(\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})-f^1(x_i)\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\middle|N\ge\frac{np}{2}\right)\ge(1-\frac{\rho}{\lfloor\frac{np}{2}\rfloor})$$

This step is perhaps not completely obvious, so let's write it out carefully. For clarity, we will temporarily denote the term $\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})-f^1(x_i)\|_\infty$ by $\|\cdot\|_\infty$. By definition of conditional probability, we have

$$\begin{aligned}\mathbb{P}\left(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\middle|N\ge\frac{np}{2}\right)&=\frac{\mathbb{P}(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\cap N\ge\frac{np}{2})}{\mathbb{P}(N\ge\frac{np}{2})}\\&=\frac{\sum_{k=\lfloor\frac{np}{2}\rfloor}^n\mathbb{P}(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\cap N=k)}{\mathbb{P}(N\ge\frac{np}{2})}\\&=\frac{\sum_{k=\lfloor\frac{np}{2}\rfloor}^n\mathbb{P}(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\Big|N=k)\mathbb{P}(N=k)}{\mathbb{P}(N\ge\frac{np}{2})}\end{aligned}$$

Now, by definition, $S^\rho_{\lfloor\frac{np}{2}\rfloor}\ge D^1_k\sqrt{\frac{k-1}{2}\ln\frac{2d_1k}{\rho}}$ for any $k\ge\lfloor\frac{np}{2}\rfloor$, and thus $\mathbb{P}(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\Big|N=k)\ge1-\frac{\rho}{k}\ge1-\frac{\rho}{\lfloor\frac{np}{2}\rfloor}$. Plugging this above, we have

$$\mathbb{P}\left(\|\cdot\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\middle|N\ge\frac{np}{2}\right)\ge(1-\frac{\rho}{\lfloor\frac{np}{2}\rfloor})\frac{\sum_{k=\lfloor\frac{np}{2}\rfloor}^n\mathbb{P}(N=k)}{\mathbb{P}(N\ge\frac{np}{2})}=1-\frac{\rho}{\lfloor\frac{np}{2}\rfloor}$$

which gives us the claim.

Now, using the law of total probability (conditioning on the random variable N), we have

$$\begin{aligned}\mathbb{P}\left(\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})-f^1(x_i)\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\right)&\ge(1-\frac{\rho}{\frac{np}{2}})\mathbb{P}(N\ge\frac{np}{2})\\&=(1-\frac{\rho}{\frac{np}{2}})(1-\mathbb{P}(N<\frac{np}{2}))\end{aligned}$$

But by a Chernoff bound 2.10, $\mathbb{P}(N<\frac{np}{2})\le\exp(-\frac{np}{8})$, so $1-\exp(-\frac{np}{8})\le1-\mathbb{P}(N<\frac{np}{2})$ and thus

$$\mathbb{P}\left(\|F^1(x_i,\{\{X_j\}\}_{v_j\in N(v_i)})-f^1(x_i)\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\right)\ge(1-\frac{\rho}{\frac{np}{2}})(1-\exp(-\frac{np}{8}))$$

Now, recall that we started the proof by conditioning on $X_i=x_i$. To get rid of this, we simply use the law of total probability (for the random variable X_i) - this is the same step as in the proof of 6.6. We obtain

$$\mathbb{P}\left(\|(S_X)^1_i-f^1(X_i)\|_\infty\le S^\rho_{\lfloor\frac{np}{2}\rfloor}\right)\ge(1-\frac{2\rho}{np})(1-\exp{(-\frac{np}{8}}))$$

□

Corollary 6.6

$\mathbb{P}\left(\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty \leq S^\rho_{\lfloor \frac{np}{2} \rfloor}\right) \geq 1 - \frac{2\rho}{p}(1-\exp(-\frac{np}{8})) - n\exp(-\frac{np}{8})$

Proof. This is just an application of the union bound. By Theorem 6.33, we know that for any node i,

$$\mathbb{P}\left(\|(S_X)_i^1 - f^1(X_i)\|_\infty > S^\rho_{\lfloor \frac{np}{2} \rfloor}\right) \leq \frac{2\rho}{np}(1-\exp(-\frac{np}{8})) + \exp(-\frac{np}{8})$$

Thus by a union bound

$$\mathbb{P}\left(\cup_{i=1}^n \|(S_X)_i^1 - f^1(X_i)\|_\infty > S^\rho_{\lfloor \frac{np}{2} \rfloor}\right) \leq \frac{2\rho}{p}(1-\exp(-\frac{np}{8})) + n\exp(-\frac{np}{8})$$

□

The above corollary is important for us, because all the terms in the bound on the right hand side tend to 0 as n goes to infinity. This was the whole point. Also observe that $S^\rho_{\lfloor \frac{np}{2} \rfloor}$ is a sub-sequence of the sequence $(S^\rho_k)_{k=1}^\infty$. By Lemma 6.32, this sequence converges to zero, and thus any subsequence also converges to 0. This allows us to prove the following key result.

Corollary 6.7

Let $G \sim \mathbb{G}(n,p)$, where $p \in (0,1]$. Then $\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty$ converges to 0 in probability.

Proof. Suppose $x > 0$ is given. We need to show that

$$\lim_{n\to\infty} \mathbb{P}(\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty \leq x) = 1$$

So suppose $\epsilon > 0$ is given. We can find ρ^* and N such that $\rho \leq \rho^*$ and $n \geq N$ imply 1 -$\frac{2\rho}{p}(1-\exp(-\frac{np}{8})) - n\exp(-\frac{np}{8}) > 1-\epsilon$. By Lemma 6.32, $S^{\rho^*}_{\lfloor \frac{np}{2} \rfloor} \to 0$ as $n \to \infty$. Thus, there is an $M \geq N$, such that $n \geq M \implies S^{\rho^*}_{\lfloor \frac{np}{2} \rfloor} < x$. Thus by Corollary 6.6, we have

$$\begin{aligned} n \geq M &\implies \mathbb{P}(\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty \leq S_n^{\rho^*}) > 1-\epsilon \\ &\implies \mathbb{P}(\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty \leq x) > 1-\epsilon \end{aligned}$$

where we used $S_n^{\rho^*} < x$. Since ϵ was arbitrary, we have $\lim_{n\to\infty} \mathbb{P}(\max_i \|(S_X)_i^1 - f^1(X_i)\|_\infty \leq x) = 1$, as desired. □

This allows us to prove a result analogous to Lemma 6.7, and obtain a zero-one law in the case for 1 layer, but now with $G \sim \mathbb{G}(n, p)$. This gives the zero-one law with respect to the E-R random graph model, in the one-layer case.

To extend the result to higher layers, recall that the proof of Theorem 6.16 uses Lemma 6.15, induction, and the McDiarmid inequality. It is clear that Lemma 6.15 remains valid even in the case of $G \sim \mathbb{G}(n, p)$, the induction base case has just been proved, and the McDiarmid inequality can be applied the same way as in Theorem 6.33. Thus indeed, the zero-one law with respect to the E-R random graph model holds even for attention models with an arbitrary number of layers.

7
Summary, critical evaluation and outlook

In this thesis we introduced a recently discovered limitation of message passing graph neural networks, discussed its connection to fundamental theorems of mathematics and discussed the expressiveness implications of this limitation. We gained an intuitive understanding of why the limitation occurs and understood the essence of the proof. The rest of the thesis focused on exploring whether this limitation occurs in attention models.

Motivated by common attention models, we defined Dirichlet-GNN, and studied its convergence behaviour. We analyzed Dirichlet random walks with sub-Gaussian random variables, where we discovered novel formulas, and successfully applied them to obtain a new zero-one law. We then used this zero-one law to formulate a conjecture about zero-one laws of attention models. We conducted experiments to verify our theoretical predictions.

Next, we considered the case of different random graph models, and observed instances when the zero-one law does not occur. We observed that the experimental setup proposed by [17] can sometimes produce surprising results, and described the need for a better experimental approach.

In the final chapter, our focus shifted to proving the zero-one law for concrete model instances, in particular Transformers, GATs and GATv2. To do this, we analyzed and carefully explained an approach presented in a very recent paper

[18], and extended the findings to the these models. We discussed some limitations concerning the random graph model and proposed a way to extend the theory to the case of graphs sampled from the E-R model.

To verify the theory, we developed a novel experimental approach for computing the continuous counterpart and its expectation for particular distributions. We observed that the behaviour observed in experiments closely matched the theoretical predictions.

Overall, the thesis achieved its goal of understanding the convergence of different model architectures, and the presence of zero-one laws in these models. However, there are several shortcomings of our work. For example, the theory presented in Chapter 6 is somewhat complicated, and we believe that it is possible to obtain a simpler theoretical framework, one that resembles the theory developed in [17] and Chapter 4. Improving and simplifying the theoretical framework is an important direction of future research.

Another shortcoming of this thesis is the lack of experiments in Chapter 6. We were not able to obtain a way to numerically approximate the continuous counterpart in different settings, which made conducting experiments with different non-linearities, model architectures or random graph models impossible.

Lastly, it is important to note that our work considered simplifications of actual models. For example, we know that the architecture of real Transformers [47] is more complicated than our transformer (Definition 2.8), and includes several attention heads and normalization layers. For this reason, our work should be thought of as a *first step* towards understanding the expressiveness of such models.

Nevertheless, our work also opened many directions for further study, which we will now discuss. An obvious direction is to study the presence of a zero-one law in other models, or in other setups. For example, one could try to eliminate the assumption that the features are sampled from a sub-Gaussian distribution, or a compact space. However, there is another interesting avenue for future research. In particular, the findings of this thesis suggest that perhaps the zero-one law is inherent to the aggregation structure of MPGNNs. We believe that this is indeed the case, and intuitively, it is so because of the laws of large numbers. Therefore, an intriguing question that naturally emerges is: Is it possible to devise an MPGNN that does not suffer from this expressiveness limitation?

Let us illustrate one way to obtain such an MPGNN. First, consider the following function on graphs

$$f(G) = \begin{cases} 1 & \textit{if \# nodes with odd degree > \# nodes with even degree} \\ 0 & \text{else} \end{cases}$$

If we sample $G \sim \mathbb{G}(n, 1/2)$, as n increases, one would expect that $f(G)$ would produce 1 around half of the time, and therefore f would not satisfy a zero-one law. This is indeed the case, but the proof of this fact is actually not that straightforward, and is omitted for brevity.

Now the key idea is the following: If we could design an MPGNN which captures the function f, then this MPGNN would not satisfy a zero-one law. Thus, to design a MPGNN which does not suffer from this expressivity limitation, we simply need to design a MPGNN which can capture f.

To do this, we need a message passing architecture that can count the number of vertices with an odd/even degree. We could design it in such a way that the first layer assigns -1 to a vertex if it has an odd degree, and +1 if it has an even degree. If we then follow with a mean pooling layer, the output of the network would be < 0 if and only if the number of odd degree nodes is greater that the number of even degree nodes. Then, all we would need is a classifier that would act as the indicator function $\mathbb{1}(x < 0)$ and f would be captured.

For the first step, we need a function which has an oscillatory behaviour depending on the parity on the input, and trigonometric functions emerge as the obvious candidates. If we assume that the initial node features are 1^d, where $d \in \mathbb{N}$ is arbitrary, we could consider the aggregation scheme given by

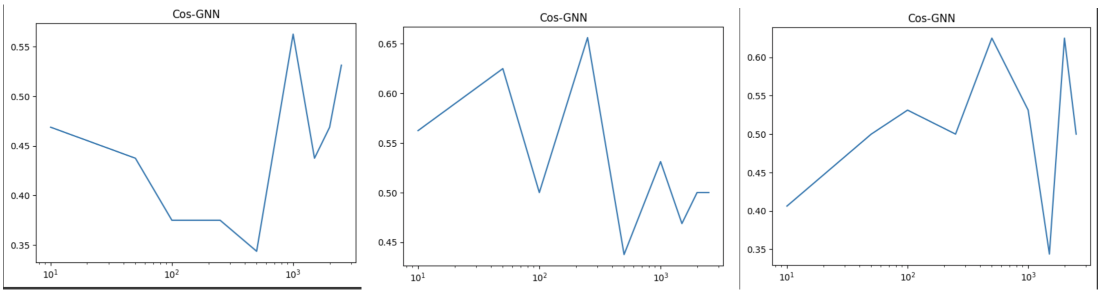
$$y_u = cos(\sum_{v \in N(u)} W x_v)$$

and parametrize $W = \frac{\pi}{d}(1^d)^T$. Then, the above becomes $y_u = cos(\pi|N(u)|)$, which means

$$y_u = \begin{cases} 1 & if\ |N(u)|\ even \\ -1 & if\ |N(u)|\ odd \end{cases}$$

which is exactly what we need. We will refer to this model as *cos-GNN*.

However, assuming that the initial features are 1^d is a strong restriction. Interestingly, experiments showed that *cosGNN* retains its oscillatory behaviour if we change the distributions of the initial node features to be uniform or normal. Furthermore, we can shift the distributions so that its mean does not land on the classification boundary, and the classifier is non-splitting. This is depicted in Figure 7.1.



(a) Cos-GNN, 1^d features **(b)** Cos-GNN, U(0,1) features **(c)** Cos-GNN, N(3,1) features

Figure 7.1: Cos-GNN does not satisfy the 0-1 law.

This suggests that in order to obtain a class of models which is strictly more expressive than all the models we've examined so far, one can consider the class of models with the aggregation scheme

$$y_u^t = \sigma\left(\sum_{v \in N(u)} \alpha_{uv} W_1^t y_v^{t-1} + cos(\sum_{v \in N(u)} W_2^t x_v) + b^t\right)$$

This class of models is clearly more expressive than attention models - if we set $W_2^t = \mathbf{0}$, we recover the class of attention models. On the other hand, a one layer model with the $W_1^1 = \mathbf{0}$ and $W = \frac{\pi}{2d}(1^d)^T$ does not satisfy a zero-one law, while most attention models do.

Future research can take multiple paths. One avenue is to investigate the performance of this class of models on conventional datasets. Alternatively, researchers might consider crafting datasets tailored specifically for this class to evaluate if they surpass the performance of traditional attention models. Additionally, one could refine the architecture described above by changing the way the non-linearity is applied, combining the attention with the *cos*, or using a different periodic function. These research directions could provide fresh insights into the capabilities of graph neural networks, and deepen our understanding of their expressive power.

Appendices

A

Appendix

We present the proof of Theorem 6.5, which is almost identical to the one in [18].

Proof Of Theorem 6.5, [18]. Plugging in the definition of a layer, we need to bound

$$\Bigg\| \sum_{j\in N(v_i)}^{n} \frac{c^{l+1}(f^l(x_i), f^l(x_j))}{\sum_{k\in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(x_k))}\psi^{l+1}(f^l(x_j)) - \\ \sum_{j\in N(v_i)}^{n} \frac{c^{l+1}(f^l(x_i), f^l(x'_j))}{\sum_{k\in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(x'_k))}\psi^{l+1}(f^l(x'_j))\Bigg\|_\infty$$

For clarity, we let

$$\begin{aligned} c_j &= c^{l+1}(f^l(x_i), f^l(x_j)) \\ c'_j &= c^{l+1}(f^l(x_i), f^l(x'_j)) \\ c &= \sum_{k\in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(x_k)) \\ c' &= \sum_{k\in N(v_i)}^{n} c^{l+1}(f^l(x_i), f^l(x'_k)) \end{aligned}$$

Now, using the triangle inequality, we have:

$$\begin{aligned} &\leq \sum_{j\in N(v_i)}^{n} \left\| \frac{c_i\psi^{l+1}(f^l(x_j))}{c} - \frac{c'_i\psi^{l+1}(f^l(x'_j))}{c'} \right\|_\infty \\ &= \sum_{j\in N(v_i)}^{n} \left\| \frac{c'c_i\psi^{l+1}(f^l(x_j)) - cc'_i\psi^{l+1}(f^l(x'_j))}{cc'} \right\|_\infty \end{aligned}$$

But by a previous lemma, we know that $\alpha < c^{l+1} \leq \beta$ for some $\alpha, \beta > 0$. Thus we have:

$$\begin{aligned}
&\leq \frac{1}{(n-1)^2\alpha^2} \sum_{j \in N(v_i)}^{n} \left\| c' c_j \psi^{l+1}(f^l(x_j)) - c c'_j \psi^{l+1}(f^l(x'_j)) \right\|_\infty \\
&= \frac{1}{(n-1)^2\alpha^2} \sum_{j \in N(v_i)}^{n} \left\| \sum_{k \in N(v_i)}^{n} c'_k c_j \psi^{l+1}(f^l(x_j)) - c_k c'_j \psi^{l+1}(f^l(x'_j)) \right\|_\infty \\
&\leq \frac{1}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \| c'_k c_j \psi^{l+1}(f^l(x_j)) - c_k c'_j \psi^{l+1}(f^l(x'_j)) \|_\infty
\end{aligned}$$

Now we can add and subtract $c'_k c'_j \psi^{l+1}(f^l(x'_j))$ to get

$$\begin{aligned}
=&\frac{1}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \| c'_k (c_j \psi^{l+1}(f^l(x_j)) - c'_j \psi^{l+1}(f^l(x'_j))) + c'_j (c'_k \psi^{l+1}(f^l(x'_j)) - \\
&c_k \psi^{l+1}(f^l(x'_j)) \|_\infty
\end{aligned}$$

Now, using the triangle inequality and the fact that $c^{l+1} \leq \beta$, we have

$$\leq \frac{\beta}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \| c_j \psi^{l+1}(f^l(x_j)) - c'_j \psi^{l+1}(f^l(x'_j)) \|_\infty + \| \psi^{l+1}(f^l(x_j))(c'_k - c_k) \|_\infty$$

Now add and subtract $c_j \psi^{l+1}(f^l(x'_j))$ in the first term, and we get

$$\begin{aligned}
&\leq \frac{\beta}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \| c_j (\psi^{l+1}(f^l(x_j)) - \psi^{l+1}(f^l(x'_j))) \|_\infty + \| \psi^{l+1}(f^l(x'_j)(c_j - c'_j) \|_\infty \\
&+ \| \psi^{l+1}(f^l(x_j))(c'_k - c_k) \|_\infty
\end{aligned}$$

Since ψ^{l+1} is linear, we know that $\psi^{l+1}(f^l(x_j)) - \psi^{l+1}(f^l(x'_j)) = \psi^{l+1}(f^l(x_j) - f^l(x'_j))$. Furthermore, for any vector x, we know that $\|\psi^{l+1}(x)\|_\infty \leq \|\psi^{l+1}\|_\infty \|x\|_\infty$ by the definition of the norm of a linear operator.

$$\begin{aligned}
&\leq \frac{\beta}{(n-1)^2\alpha^2} \sum_{j,k \in N(v_i)}^{n} \beta \|\psi\|_\infty \|f^l(x_j) - f^l(x'_j)\|_\infty + \|\psi\|_\infty \|f^l(x'_j)\|_\infty |c_j - c'_j| \\
&\qquad\qquad + \|\psi\|_\infty \|f^l(x'_j)\|_\infty |c_k - c'_k| \\
&\leq \frac{\beta}{(n-1)^2\alpha^2} \Big((n-1)\beta \|\psi\|_\infty 2 \|f^l\|_\infty + (n-1) 4\beta |\psi\|_\infty \|f^l\|_\infty \Big) \\
= &\frac{\beta^2 (6 \|\psi\|_\infty \|f^l\|_\infty)}{(n-1)\alpha^2} = O(1/n)
\end{aligned}$$

as required. □

References

[1] Wenqi et al. “Graph Neural Networks for Social Recommendation”. In: *World Wide Web Conference*. 2019.

[2] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. “Modeling Polypharmacy Side Effects with Graph Convolutional Networks”. In: (May 2018). DOI: 10.1101/258814.

[3] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. “Graph neural networks in particle physics”. In: *Machine Learning: Science and Technology* 2.2 (Jan. 2021), p. 021001. DOI: 10.1088/2632-2153/abbf9a. URL: https://doi.org/10.1088%2F2632-2153%2Fabbf9a.

[4] Yifei Shen et al. “Graph Neural Networks for Wireless Communications: From Theory to Practice”. In: *Trans. Wireless. Comm.* 22.5 (Nov. 2022), pp. 3554–3569. ISSN: 1536-1276. DOI: 10.1109/TWC.2022.3219840. URL: https://doi.org/10.1109/TWC.2022.3219840.

[5] M. Gori, G. Monfardini, and F. Scarselli. “A new model for learning in graph domains”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. 2005, 729–734 vol. 2. DOI: 10.1109/IJCNN.2005.1555942.

[6] Franco Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[7] Peter Battaglia et al. “Relational inductive biases, deep learning, and graph networks”. In: *arXiv* (2018). URL: https://arxiv.org/pdf/1806.01261.pdf.

[8] Jie Zhou et al. “Graph neural networks: A review of methods and applications”. In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012.

[9] Alex Fout et al. “Protein Interface Prediction using Graph Convolutional Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f507783927f2ec2737ba40afb
Paper.pdf.

[10] Reau et al. “DeepRank-GNN: a graph neural network framework to learn patterns in protein-protein interfaces”. In: *Bioinformatics* (2022).

[11] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[12] Daniel Cummings and Marcel Nassar. "Structured Citation Trend Prediction Using Graph Neural Networks". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020. DOI: 10.1109/icassp40776.2020.9054769. URL: https://doi.org/10.1109%2Ficassp40776.2020.9054769.

[13] Shiwen Wu et al. *Graph Neural Networks in Recommender Systems: A Survey*. 2022. arXiv: 2011.02260 [cs.IR].

[14] David K Duvenaud et al. "Convolutional Networks on Graphs for Learning Molecular Fingerprints". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf.

[15] Hannes Stärk et al. "3D Infomax improves GNNs for Molecular Property Prediction". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 20479–20502. URL: https://proceedings.mlr.press/v162/stark22a.html.

[16] Patrick Reiser et al. "Graph neural networks for materials science and chemistry". In: *Communications Materials* 3.1 (2022), p. 93. DOI: 10.1038/s43246-022-00315-6.

[17] Sam Adam-Day, Theodor Mihai Iliant, and İsmail İlkan Ceylan. *Zero-One Laws of Graph Neural Networks*. 2023. arXiv: 2301.13060 [cs.LG].

[18] Matthieu Cordonnier et al. *Convergence of Message Passing Graph Neural Networks with Generic Aggregation On Large Random Graphs*. 2023. arXiv: 2304.11140 [stat.ML].

[19] Justin Gilmer et al. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 1263–1272.

[20] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[21] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[22] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML].

[23] Shaked Brody, Uri Alon, and Eran Yahav. *How Attentive are Graph Attention Networks?* 2022. arXiv: 2105.14491 [cs.LG].

[24] *Sub-Gaussian random variables.*

[25] https://ocw.mit.edu/courses/18-s997-high-dimensional-statistics-spring-2015/a69e2f53bb2eeb

[26] Keyulu Xu et al. “How Powerful are Graph Neural Networks?” In: *ICLR*. 2019.

[27] Christopher Morris et al. *Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks*. 2021. arXiv: 1810.02244 [cs.LG].

[28] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. “Random Features Strengthen Graph Neural Networks”. In: *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Virtual Event, April 29 - May 1, 2021*. Ed. by Carlotta Demeniconi and Ian Davidson. SIAM, 2021, pp. 333–341. DOI: 10.1137/1.9781611976700.38. URL: https://doi.org/10.1137/1.9781611976700.38.

[29] R Abboud et al. “The surprising power of graph neural networks with random node initialization”. In: International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 2112–2118.

[30] Qimai Li, Zhichao Han, and Xiao-ming Wu. “Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11604. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11604.

[31] Uri Alon and Eran Yahav. *On the Bottleneck of Graph Neural Networks and its Practical Implications*. 2021. arXiv: 2006.05205 [cs.LG].

[32] Jake Topping et al. *Understanding over-squashing and bottlenecks on graphs via curvature*. 2022. arXiv: 2111.14522 [stat.ML].

[33] Keyulu Xu et al. *Representation Learning on Graphs with Jumping Knowledge Networks*. 2018. arXiv: 1806.03536 [cs.LG].

[34] Kenta Oono and Taiji Suzuki. *Graph Neural Networks Exponentially Lose Expressive Power for Node Classification*. 2021. arXiv: 1905.10947 [cs.LG].

[35] Ralph Abboud et al. “The Surprising Power of Graph Neural Networks with Random Node Initialization”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Ed. by Zhi-Hua Zhou. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 2112–2118. DOI: 10.24963/ijcai.2021/291. URL: https://doi.org/10.24963/ijcai.2021/291.

[36] Pablo Barceló et al. “The logical expressiveness of graph neural networks”. In: *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethopia, April 26–30, 2020*. 2020.

[37] Martin Grohe. *The Descriptive Complexity of Graph Neural Networks*. 2023. arXiv: 2303.04613 [cs.LO].

[38] Ronald Fagin. “Probabilities on Finite Models”. In: *The Journal of Symbolic Logic* 41.1 (1976), pp. 50–58. ISSN: 00224812. URL: http://www.jstor.org/stable/2272945.

[39] *Zero-One Laws for Random Graphs.* https://jeremykun.com/2015/02/09/zero-one-laws-for-random-graphs/.

[40] *Rado graph.* https://en.wikipedia.org/wiki/Rado_graph.

[41] A. Erdős P.; Rényi. “"Asymmetric graphs"”. In: *Acta Mathematica Academiae Scientiarum Hungarica* 14.3 (1963), pp. 295–315.

[42] Gerard Letac and Mauro Piccioni. “Dirichlet random walks”. In: *Journal of Applied Probability* 51.4 (2014), pp. 1081–1099.

[43] Nasrollah Etemadi. “Convergence of Weighted Averages of Random Variables Revisited”. In: *Proceedings of the American Mathematical Society* 134.9 (2006), pp. 2739–2744. ISSN: 00029939, 10886826. URL: http://www.jstor.org/stable/4098124 (visited on 08/17/2023).

[44] Joel Spencer. *The Strange Logic of Random Graphs.* Springer Berlin, Heidelberg, 2001.

[45] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. “On the Universality of Graph Neural Networks on Large Random Graphs”. In: *Advances in Neural Information Processing Systems.* Ed. by A. Beygelzimer et al. 2021. URL: https://openreview.net/forum?id=Xci6vUAGeJ.

[46] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. “Convergence and Stability of Graph Convolutional Networks on Large Random Graphs”. In: *Advances in Neural Information Processing Systems.* Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21512–21523. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/f5a14d4963acf488e3a24780a84ac96c-Paper.pdf.

[47] Ashish Vaswani et al. *Attention Is All You Need.* 2023. arXiv: 1706.03762 [cs.CL].