

# GRAPH NEURAL NETWORKS ARE MORE POWERFUL THAN WE THINK

Charilaos I. Kanatsoulis, and Alejandro Ribeiro

Dept. of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

## ABSTRACT

Graph Neural Networks (GNNs) are powerful architectures that have demonstrated remarkable performance in various node-level and graph-level tasks. Despite this success, prominent analysis shows that their representation power is limited and that they are at most as expressive as the Weisfeiler-Lehman (WL) test. In this paper, we take a different approach and analyze the expressive power of GNNs with respect to the spectral decomposition of the graph operators. We prove that GNNs can produce distinct equivariant outputs for all graphs with different eigenvalues, therefore surpassing the limitations of the WL test. On the practical front, our approach enables the design of GNNs that unlock the full potential of their expressive power. Thorough experimental analysis on graph classification datasets supports our theoretical findings and showcases the effectiveness of the proposed approach.

**Index Terms**— Graph Neural Networks, representation power, spectral analysis, Weisfeiler-Lehman test, permutation equivariance

## 1. INTRODUCTION

The success of Graph Neural Networks (GNNs) across a wide range of tasks has sparked significant research interest in demystifying their functionality and understanding their fundamental properties. In this pursuit, [1] delves into the analysis of GNNs' permutation invariance-equivariance, and [2] proves their stability to perturbations. An important line of work investigates the transferability of GNNs to very large graphs [3, 4], while [5, 6] discuss the universality properties of GNNs, particularly in the context of functions that exhibit permutation invariance or equivariance.

Despite the theoretical and empirical evidence for the success of GNNs, the current analysis shows that their representation power is limited. In particular, the prominent works of [7, 8] study the expressive power of local, message-passing GNNs with anonymous countable inputs and show that they are at most as powerful as the Weisfeiler-Lehman (WL) test [9]. To overcome these limitations, several follow-up works have been proposed that can be grouped into three categories. The first set of methods [10–15] designs more complex architectures, that use  $k$ -tuple and  $k$ -subgraph information in the form of a tensor. These works employ more expressive structures compared to standard GNNs, however, they are usually computationally heavier to implement and also prone to overfitting. The second group of methods [16–18] compute features related to the subgraph information, to increase the separation capabilities of GNNs. Subgraph attributes benefit various tasks, however, increased expressivity results from independent graph algorithms and not from the GNN architecture. The third set of methods, [19, 20], uses random features as input to the GNN in order to achieve enhanced function approximation. However, the increased function approximation comes at the expense of permutation equivariance and leads to poor generalization.

In this paper, we revisit local, message-passing GNNs and show that they are more powerful than initially thought. In particular, we prove that the local, message-passing operations are able to generate equivariant representations, from anonymous inputs, that are more expressive than the WL test. To be more precise, our work is motivated by the following research problem:

**Problem definition:** Given a pair of different graphs  $\mathcal{G}, \hat{\mathcal{G}}$  and anonymous inputs  $\mathbf{X}, \hat{\mathbf{X}}$ ; is there an equivariant GNN  $\phi$  with parameter tensor  $\mathcal{H}$  such that  $\phi(\mathbf{X}; \mathcal{G}, \mathcal{H}), \phi(\hat{\mathbf{X}}; \hat{\mathcal{G}}, \mathcal{H})$  are nonisomorphic?

Anonymous inputs are defined as the inputs that are structure- and identity-agnostic, i.e., they carry zero information at all. The study of anonymous inputs stems from our interest in examining the ability of GNNs to generate information about the graph structure when zero information is given in the input.

In this paper, we give an affirmative answer to the aforementioned research question. In particular, we analyze message-passing GNNs with white random inputs and prove that they can generate discriminative outputs, for at least all graphs with different eigenvalues. Consequently, message-passing GNNs are more expressive than the WL test as they can discriminate between graphs that fail the WL test, yet have different eigenvalues. Note that, having different eigenvalues is a very mild condition that most real graphs satisfy. Our analysis employs spectral decomposition tools to analyze the representation power of GNNs and show that the source of the WL test as a limit is not the message-passing mechanism but the use of the all-one input.

On the practical front, our work maintains permutation equivariance, a fundamental property that is abandoned in related works with random input features. Furthermore, the proposed analysis is constructive and enables the design of efficient GNN architectures that can potentially unlock the full potential of message-passing operations. To avoid training GNNs with random input, we prove that the whole process is equivalent to a deterministic GNN that in the first layer linearly combines the number of closed paths each node participates in. Finally, numerical experiments with graph isomorphism and graph classification datasets showcase that the proposed GNN approach demonstrates remarkable performance and is a better anonymous discriminator compared to WL-related approaches.

**Other related works:** The first study into the approximation capabilities of GNNs was conducted by [21]. The works of [22] and [23], leverage multiple and directional aggregators to enhance the expressiveness of GNNs. On the other hand, [24, 25] introduces convolutional operations in the spectral domain to develop powerful GNNs. Additionally, [26] examines a GNN's learning capacity concerning its width and depth, while [27] establishes a link between universal approximation and the capacity of GNNs.

**Notation:** Lowercase bold letters,  $\mathbf{x}$ , represent vectors, uppercase bold letters,  $\mathbf{X}$ , represent matrices, and calligraphic uppercase letters,  $\mathcal{X}$ , denote sets. For a matrix  $\mathbf{X}$  or vector  $\mathbf{x}$ , we denote their transpose by  $\mathbf{X}^T$  and  $\mathbf{x}^T$ , respectively, and  $\text{diag}[\mathbf{X}]$  denotes the vector that contains the diagonal of matrix  $\mathbf{X}$ .

Emails: kanac@seas.upenn.edu, aribeiro@seas.upenn.edu

## 2. PRELIMINARIES

A graph is represented by  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ , with a set of vertices  $\mathcal{V} = \{1, \dots, N\}$ , and a set of edges  $\mathcal{E} = \{(v, u)\}$ , or equivalently by an adjacency matrix  $\mathbf{S} \in \{0, 1\}^{N \times N}$ . The vertices (nodes) of the graph are often associated with graph signals  $\mathbf{x}_v \in \mathbb{R}^D$  with  $D$  features, also known as node attributes. We study undirected graphs, thus  $\mathbf{S}$  is symmetric and admits a spectral decomposition  $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ , where  $\mathbf{U}$  is an orthonormal matrix containing the eigenvectors, and  $\mathbf{\Lambda}$  is a diagonal matrix of the corresponding eigenvalues.

A pair of graphs  $\mathcal{G}, \hat{\mathcal{G}}$  is considered nonisomorphic if and only if there is no permutation matrix  $\mathbf{\Pi}$  such that  $\mathbf{S} = \mathbf{\Pi}\hat{\mathbf{S}}\mathbf{\Pi}^T$ . Our analysis studies a broad class of nonisomorphic graphs that have different eigenvalues. To be more precise, let  $\mathcal{S}, \hat{\mathcal{S}}$  be the set containing the unique eigenvalues of  $\mathbf{S}$  and  $\hat{\mathbf{S}}$  with multiplicities denoted by  $m_\lambda, \hat{m}_\lambda$  respectively.

**Assumption 2.1.**  $\mathcal{S}, \hat{\mathcal{S}}$  have different eigenvalues, i.e., there exists  $\lambda \in \mathcal{S}$ , such that  $\lambda \notin \hat{\mathcal{S}}$  or  $m_\lambda \neq \hat{m}_\lambda$ .

Assumption 2.1 is not restrictive. Real nonisomorphic graphs have different eigenvalues with very high probability [28].

We study standard GNNs that perform local, message-passing operations. A message-passing GNN is a cascade of layers and is defined by the following recursive formula:

$$\mathbf{x}_v^{(r)} = g^{(r-1)} \left( \mathbf{x}_v^{(r-1)}, f^{(r-1)} \left( \left\{ \mathbf{x}_u^{(r-1)} : u \in \mathcal{N}(v) \right\} \right) \right), \quad (1)$$

where  $\mathcal{N}(v)$  is the neighborhood of vertex  $v$ , i.e.,  $u \in \mathcal{N}(v)$  if and only if  $(u, v) \in \mathcal{E}$ . The function  $f^{(r)}$  aggregates information from the multiset of neighboring signals, whereas  $g^{(r)}$  combines the signal of each vertex with the aggregated one from the neighboring vertices. Common choices for  $f^{(r)}, g^{(r)}$  are the multi-layer perceptron (MLP), the linear function, and the summation function. In this paper, we study the following message-passing GNN:

$$\mathbf{X}^{(l)} = \rho \left[ \sigma \left( \sum_{k=0}^{K-1} \mathbf{S}^k \mathbf{X}^{(l-1)} \mathbf{H}_k^{(l-1)} \right) \right], \quad (2)$$

where  $\mathbf{X} = \mathbf{X}^{(0)} \in \mathbb{R}^{N \times D}$  represents the signals of all vertices,  $\mathbf{H}_k^{(l)} \in \mathbb{R}^{D \times F_l}$  are the trainable parameters of layer  $l$ ,  $\sigma$  is a pointwise nonlinear activation function, and  $\rho$  is a normalization function. The layer in (2) is equivalent to  $K-1$  layers in (1), when  $f^{(r)}$  is the summation function, and  $g^{(r)}$  is the multivariate linear function for the first  $K-2$  layers and the MLP followed by a normalization function for the  $K$ -th layer.

## 3. LIMITATIONS OF WL-RELATED ANALYSIS

The analyses in [7, 8] study the expressive power of message-passing GNNs by following the WL test steps. In both analyses, there is an implicit assumption that the input to the GNN is  $\mathbf{x} = \mathbf{1}$ . This assumption originates from the first step of the WL test, which assigns an identical color to the representation of every node. As a result, [7, 8], study the expressive power of GNNs with the all-one input  $\phi(\mathbf{1}; \mathcal{G}, \mathcal{H})$ , instead of the expressive power of GNNs with general inputs  $\phi(\mathbf{X}; \mathcal{G}, \mathcal{H})$ . Studying a GNN with  $\mathbf{x} = \mathbf{1}$  is limiting for two reasons. First, the analysis is constrained to only a subset of possible inputs (the constant inputs) and cannot derive concrete conclusions about the whole set of possible inputs. Second, the all-one input is associated with serious limitations, involving the spectral decomposition of the graph.

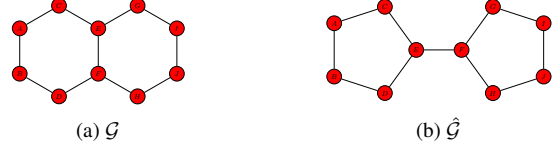


Fig. 1: WL indistinguishable graphs

To analyze the spectral limitations of the all-one input we first observe that for a single-feature input, the GNN layer in (2) takes the form  $\mathbf{X}^{(1)} = \rho \left[ \sigma \left( \sum_{k=0}^{K-1} \mathbf{S}^k \mathbf{x} \mathbf{h}_k^{(0)T} \right) \right]$ , which is a set of  $F_1$  parallel graph filters of the form:

$$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}, \quad (3)$$

followed by the pointwise nonlinearity and the normalization function. The expression in (3) is also known as the graph convolution and can also be cast as:

$$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{U} \mathbf{\Lambda}^k \mathbf{U}^T \mathbf{x} = \sum_{n=1}^N \tilde{h}(\lambda_n) \tilde{x}_n \mathbf{u}_n, \quad (4)$$

where  $\tilde{h}(\lambda_n) = \sum_{k=0}^{K-1} h_k \lambda_n^k$  is the  $n$ -th component of the Graph Fourier Transform (GFT) [29] of the graph filter with  $\{h_k\}_{k=0}^{K-1}$  parameters, and  $\tilde{x}_n = \mathbf{u}_n^T \mathbf{x}$  is the  $n$ -th component of the GFT of  $\mathbf{x}$ . We observe that the graph convolution is a linear combination of the eigenvectors of  $\mathbf{S}$  weighted by the frequency response of the filter, and the GFT of  $\mathbf{x}$ . For  $\mathbf{x} = \mathbf{1}$ ,  $\tilde{x}_n = \mathbf{u}_n^T \mathbf{1}$  and  $\tilde{x}_n = 0$  when  $\mathbf{u}_n$  is orthogonal to  $\mathbf{1}$ . As a result, when using an all-one input, it will effectively suppress all spectral components (eigenvectors) of the graph that are orthogonal to  $\mathbf{1}$ . This can lead to the loss of crucial information about the graph's structure. This limitation becomes even more significant when considering that the majority of real-world graphs possess spectral components orthogonal to  $\mathbf{1}$ . It is important to note that while the pointwise nonlinearity may potentially recover the dismissed components, in the case of 1-WL indistinguishable graphs, these spectral components will remain zero.

A classic pair of graphs with different eigenvalues, that the WL algorithm and GNNs with  $\mathbf{x} = \mathbf{1}$  input cannot tell apart (1-WL indistinguishable graphs), is presented in Fig. 1. The eigenvalues along with the GFT of  $\mathbf{x} = \mathbf{1}$  for these graphs is presented in Table 1. We

Table 1: Eigenvalue and GFT information for the graphs in Fig. 1.

GRAPH		$n$									
		1	2	3	4	5	6	7	8	9	10
$\mathcal{G}$	$\lambda_n$	2.3	1.6	1.3	1.0	0.6	-2.3	-1.6	-0.6	-1.0	-1.3
	$\mathbf{u}_n^T \mathbf{1}$	3.0	0	0	-0.8	0	0	0	0	0	-0.2
$\hat{\mathcal{G}}$	$\hat{\lambda}_n$	2.3	1.9	1.0	0.6	0.6	0.3	-1.3	-1.6	-1.6	-2.1
	$\hat{\mathbf{u}}_n^T \mathbf{1}$	3.0	0	-0.8	0	0	0	-0.2	0	0	0

observe that  $\mathbf{x} = \mathbf{1}$  is orthogonal to the eigenvectors corresponding to the eigenvalues that differentiate the two graphs. Consequently, the operation  $\mathbf{S} \cdot \mathbf{1}$  negates the vital information that tells these two graphs apart and the WL test or message-passing GNNs with  $\mathbf{x} = \mathbf{1}$  input fail to tell these graphs apart.

## 4. ANALYZING GNNs WITH RANDOM NOISE

To overcome the limitations of the WL-related analysis, we study message-passing GNNs with white noise. In particular, we consider a

message-passing GNN layer in (2) which we load with white random input  $\mathbf{x} \in \mathbb{R}^N$ , i.e.,  $\mathbb{E}[\mathbf{x}] = 0$ ,  $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \sigma^2 \mathbf{I}$ . Since the input is a vector the GNN layer in (2) corresponds to a set of  $F_1$  separable operations of type:

$$\mathbf{y} = \rho[\sigma(\mathbf{z})] = \rho\left[\sigma\left(\sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}\right)\right] \quad (5)$$

First, we study the effect of the white input to the linear part of (5), i.e., the graph filter (as defined in (3)). Since  $\mathbf{x}$  is a zero-mean random vector,  $\mathbf{z}$  is also a random vector with  $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ . Thus, the expected value provides no information about the graph structure. Measuring the variance, on the other hand, yields:

$$\begin{aligned} \text{var}[\mathbf{z}] &= \mathbb{E}[\mathbf{z}^2] = \text{diag}\left[\mathbb{E}[\mathbf{z}\mathbf{z}^T]\right] \\ &= \text{diag}\left[\sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbb{E}[\mathbf{x}\mathbf{x}^T] \sum_{m=0}^{K-1} h_m \mathbf{S}^m\right] \\ &= \sigma^2 \sum_{k=0}^{K-1} \sum_{m=0}^{K-1} h_k h_m \text{diag}[\mathbf{S}^k \mathbf{S}^m] = \sum_{k=0}^{2K-2} h'_k \text{diag}[\mathbf{S}^k], \quad (6) \end{aligned}$$

where  $h'_k = \sigma^2 \sum_{m,l} h_m h_l$ , such that  $m + l = k$ . The results of Equation (6) provide two notable points. First, the variance of a graph filter with white random input is a linear combination of powers of the adjacency diagonals. In other words, the variance linearly combines the  $k$ -hop closed loops each node is involved in. Furthermore, the resulting output aggregates information from  $(2K - 1)$ -hop neighborhood, whereas the original filter has length  $K$ . This is attributed to the square nonlinearity introduced by the variance computation. However, there is a caveat that the degrees of freedom for  $\{h'_k\}_{k=0}^{2K-2}$  are  $K$  and not  $2K - 1$ .

The most important outcome from the above analysis is that the variance of the graph filter with white input is a proper message-passing layer, as defined in (5). To be more precise, the message-passing block in (5) with white input, square pointwise nonlinearity, and normalization defined by the expectation operator computes the variance of the graph filter output, which produces a permutation equivariant representation for the nodes of the graph that linearly combine of  $k$ -hop closed loops, i.e.,

$$\mathbf{y} = \rho\left[\sigma\left(\sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}\right)\right] = \sum_{k=0}^{2K-2} h'_k \text{diag}[\mathbf{S}^k], \quad (7)$$

where  $\sigma(\cdot) = (\cdot)^2$ , and  $\rho[\cdot] = \mathbb{E}[\cdot]$ . The above analysis proves the following theorem.

**Theorem 4.1.** *A message-passing GNN  $\phi(\mathbf{X}; \mathcal{G}, \mathcal{H})$ , defined by (7), with white random input produces equivariant representations that can count all the closed paths each node participates in.*

To analyze the expressive power of message-passing GNNs, we study the module (7) with spectral decomposition tools:

$$\begin{aligned} \mathbf{y} &= \sum_{k=0}^{2K-2} h_k \text{diag}\left(\sum_{n=1}^N \lambda_n^k \mathbf{u}_n \mathbf{u}_n^T\right) \\ &= \sum_{k=0}^{2K-2} \sum_{n=1}^N h_k \lambda_n^k |\mathbf{u}_n|^2 = \sum_{n=1}^N \tilde{h}(\lambda_n) |\mathbf{u}_n|^2, \quad (8) \end{aligned}$$

where  $\tilde{h}(\lambda_n) = \sum_{k=0}^{2K-2} h_k \lambda_n^k$  is the frequency response of the graph filter in (7) at  $\lambda_n$ . In simple words, the output of the message

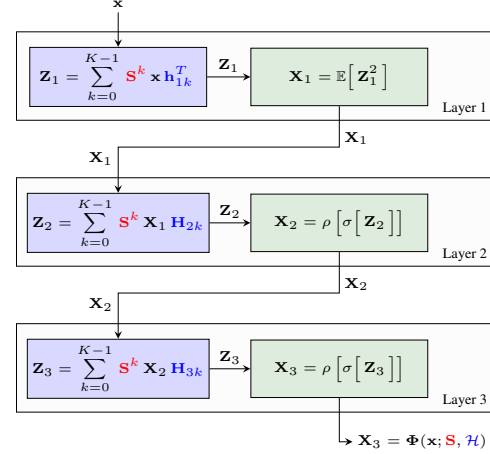


Fig. 2: Proposed Architecture

passing block in (7) is a linear combination of the absolute values of the graph adjacency eigenvectors weighted by the frequency response of the filter parameters.

**Corollary 4.2.** *The  $k$ -hop paths in a graph are associated with the eigenvalue decomposition of the graph adjacency as follows:*

$$\mathbf{d}^k := \text{diag}(\mathbf{S}^k) = \sum_{n=1}^N \lambda_n^k |\mathbf{u}_n|^2. \quad (9)$$

Following the above analysis, we characterize the expressive power of message-passing GNNs as follows:

**Theorem 4.3.** *Let  $\mathcal{G}$ ,  $\hat{\mathcal{G}}$  be nonisomorphic graphs. If Assumption 2.1 holds, there exists a GNN with white random input that produces equivariant nonisomorphic representations for the two graphs.*

The proof is relegated to the journal version due to space limitations. The implications of Theorem 4.3 are noteworthy. A GNN with white input can uniquely process the majority of real graphs, i.e., those that have different eigenvalues. Furthermore, although the input is random, the expectation normalization produces deterministic outputs that always maintain permutation equivariance. This is a fundamental difference between our work and those in [19,20], which prove that random inputs produce expressive representations but not equivariant ones. Note that expressivity and permutation equivariance are both fundamental properties in deep learning. Our previous analysis shows that GNNs with random inputs are both expressive and equivariant, and the key is treating white noise as anonymous input to the GNN and not just drawing unique random realizations.

## 5. ARCHITECTURE

To transform the previous analysis to a multi-layer GNN we propose the architecture shown in Fig. 2. The first layer is loaded with white Gaussian noise which is processed by a set of message-passing blocks defined in (7). In practice, we do not feed the GNN with random noise but train the equivalent model instead, which linearly combines the  $k$ -hop loops. The remaining layers are designed according to Equation (2). Comparing the proposed architecture with the WL-related GNNs we get the following:

**Corollary 5.1.** *The proposed architecture in Fig. 2 is strictly more expressive compared to GNNs with  $\mathbf{x} = \mathbf{1}$  or  $\mathbf{x} = \mathbf{S} \cdot \mathbf{1}$  inputs.*

**Table 2:** Average testing score and standard deviation over 10 shuffles

Dataset	GNN <sub>random</sub>		GIN <sub>1</sub>		GIN <sub>random</sub>	
	micro F1	macro F1	micro F1	macro F1	micro F1	macro F1
CSL	<b>100 ± 0</b>	<b>100 ± 0</b>	10 ± 3.3	1.8 ± 0.6	<b>100 ± 0</b>	<b>100 ± 0</b>
IMDBBINARY	71.7 ± 2.5	71.3 ± 2.7	<b>74.7 ± 3.2</b>	<b>74.6 ± 3.2</b>	<b>71.6 ± 3.4</b>	<b>71 ± 3.8</b>
IMDBMULTI	46.1 ± 2.8	44.2 ± 3.2	<b>50.3 ± 2.8</b>	<b>48 ± 3.4</b>	48.6 ± 2.9	46.1 ± 4.2
REDDITBINARY	87.2 ± 4.1	87.1 ± 4.3	81.6 ± 5.6	81.5 ± 5.7	<b>89.8 ± 2.3</b>	<b>89.7 ± 2.3</b>
REDDITMULTI	54 ± 2.2	52.4 ± 2.1	52.4 ± 2.4	50.9 ± 2.4	<b>55 ± 1.5</b>	<b>53.6 ± 1.7</b>
PTC	63.6 ± 4.9	61.4 ± 6.9	<b>65.7 ± 8.8</b>	<b>65.1 ± 9.1</b>	62.5 ± 5.1	61.4 ± 5.5
PROTEINS	74.2 ± 4.2	73 ± 4	74 ± 4.6	72.3 ± 4.5	<b>74.3 ± 4.8</b>	<b>73.1 ± 4.5</b>
MUTAG	89.3 ± 7.3	87.2 ± 9.3	<b>89.8 ± 7.6</b>	88.6 ± 8.8	<b>89.8 ± 8</b>	<b>88.7 ± 8.6</b>
NCI1	74.5 ± 2.1	74.3 ± 2.1	<b>77.2 ± 1.9</b>	<b>77.2 ± 1.9</b>	<b>76.3 ± 3.7</b>	<b>76.2 ± 3.8</b>

To give a concrete example. An 1-layer GNN version of Fig. 2 with parameters  $(h_0, h_1, h_2, h_3, h_4, h_5) = (0, 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \frac{1}{5})$  produces different representations for the graphs in Fig. 1, that fail the WL test.

## 6. EXPERIMENTS

In this section we evaluate the performance of the proposed analysis with graph classification datasets. In particular, we study the effect of using anonymous all-one inputs vs anonymous random inputs on the expressivity of GNNs. To this end, three architectures are considered: the GNN of Fig. 2, denoted as GNN<sub>random</sub>, the Graph Isomorphism Network (GIN) [8] with all-one input, denoted as GIN<sub>1</sub>, and the GIN augmented with the first layer of Fig. 2, denoted as GIN<sub>random</sub>.

### 6.1. The CSL dataset

The first experiment is performed on the Circular Skip Link (CSL) dataset [11]. The CSL dataset is widely regarded as the benchmark standard for evaluating GNNs concerning isomorphism [30]. It contains of 150 4-regular graphs, each of which consists of 41 nodes and 164 edges, classified into 10 distinct categories.

Since the CSL graphs are regular,  $\mathbf{x} = \mathbf{1}$  is always an eigenvector and orthogonal to all other eigenvectors. Consequently,  $\mathbf{x} = \mathbf{1}$  dismisses all the valuable information of the graphs, and as indicated in Table 2, the classification performance is completely random. In contrast, the proposed GNN architectures handle this dataset adeptly. Specifically, a single layer GNN<sub>random</sub>, characterized by parameters  $(h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9) = (0, 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \frac{1}{5}, -\frac{1}{6}, \frac{1}{7}, -\frac{1}{8}, \frac{1}{9})$  can classify these graphs with a remarkable accuracy rate of 100%. To illustrate this, in Table 3 we present the normalized GNN output (summed over all nodes)  $\mathbf{1}^T \mathbf{y} / 10^3$  for each class of graphs in the CSL dataset. The output remains identical for graphs within the same class but differs for graphs belonging to distinct classes. As a result, achieving perfect classification accuracy boils down to passing the GNN output to a simple linear classifier or even a linear assignment algorithm.

**Table 3:** GNN output  $\mathbf{1}^T \mathbf{y} / 10^3$  for CSL classes.

CLASS									
0	1	2	3	4	5	6	7	8	9
74	-46	0.1	-31	-25	-26	-18	-29	16	-21

### 6.2. Social and biological networks

Next, the performance of the three architectures is tested with standard social, chemical, and bioinformatics graph classification datasets

[31]. The dataset statistics can be found in Table 4. Each GNN model is designed with 4 layers, including skip-connections, and the same number of neurons. The output of each layer is processed by a graph pooling layer and then passed to a linear classifier. The used nonlinearity is ReLU. To test the performance of the three architectures each dataset is divided into 50 – 50 training-testing splits and 10-fold cross-validation is performed. The performance is assessed by measuring the micro F1 and macro F1 score for each epoch and presenting the epoch with the best average result among the 10 folds. The testing results over 10 shuffles are presented in Table 2.

**Table 4:** Datasets

Dataset	Graphs	Vertices	Edges	Classes	Network Type
CSL	150	41	164	10	Circulant
IMDBBINARY	1,000	20	193	2	Social
IMDBMULTI	1,500	13	132	3	Social
REDDITBINARY	2000	430	498	2	Social
REDDITMULTI	5000	509	595	5	Social
PTC	344	26	52	3	Bioinformatic
PROTEINS	1,113	39	146	2	Bioinformatic
MUTAG	188	18	20	2	Chemical
NCI1	4110	39	73	2	Chemical

Table 2 reveals that GIN<sub>1</sub> experiences a significant performance gap when compared to the GNN<sub>random</sub> architecture and GIN<sub>random</sub> in the REDDITBINARY dataset, while notable improvements are evident in the REDDITMULTI dataset. Conversely, GIN<sub>1</sub> exhibits a modest 3% advantage in the IMDBBINARY dataset, whereas, in the remaining datasets, the performances of the competing algorithms are statistically the same. The latter can be attributed to the fact that the vital classification components, of these datasets, are not orthogonal to  $\mathbf{x} = \mathbf{1}$  and GIN<sub>1</sub> is not undergoing critical information loss. In summary, we conclude that GNN<sub>random</sub> and GIN<sub>random</sub> do not only show remarkable classification performance in most datasets but can also handle pathological datasets such as the CSL. This underscores the importance of thoroughly analyzing the representation capabilities of GNNs and tailoring GNN architectures accordingly.

## 7. CONCLUSION

In this paper, we studied the expressive power of message-passing GNNs with random noise and spectral decomposition tools. We proved that contrary to the current analysis, the WL test is not the real expressivity limit and proved that GNNs can produce discriminative and equivariant representations for any pair of graphs with different eigenvalues. Following this analysis, we designed powerful architectures that demonstrate remarkable performance with graph isomorphism and graph classification datasets.

## 8. REFERENCES

- [1] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman, "Invariant and equivariant graph networks," in *International Conference on Learning Representations*, 2018.
- [2] Fernando Gama, Joan Bruna, and Alejandro Ribeiro, "Stability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5680–5695, 2020.
- [3] Luana Ruiz, Luiz Chamon, and Alejandro Ribeiro, "Graphon neural networks and the transferability of graph neural networks," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1702–1712.
- [4] Ron Levie, Wei Huang, Lorenzo Bucci, Michael Bronstein, and Gitta Kutyniok, "Transferability of spectral graph convolutional neural networks," *Journal of Machine Learning Research*, vol. 22, no. 272, pp. 1–59, 2021.
- [5] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman, "On the universality of invariant networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4363–4371.
- [6] Nicolas Keriven and Gabriel Peyré, "Universal invariant and equivariant graph neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [7] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe, "Weisfeiler and leman go neural: higher-order graph neural networks," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019, pp. 4602–4609.
- [8] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," in *International Conference on Learning Representations*, 2019.
- [9] Boris Weisfeiler and Andrei Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *NTI Series*, vol. 2, no. 9, pp. 12–16, 1968.
- [10] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman, "Provably powerful graph networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [11] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro, "Relational pooling for graph representations," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4663–4673.
- [12] Waiss Azizian et al., "Expressive power of invariant and equivariant graph neural networks," in *International Conference on Learning Representations*, 2020.
- [13] Christopher Morris, Gaurav Rattan, and Petra Mutzel, "Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21824–21840, 2020.
- [14] Floris Geerts and Juan L Reutter, "Expressiveness and approximation properties of graph neural networks," in *International Conference on Learning Representations*, 2021.
- [15] Lorenzo Giusti, Claudio Battiloro, Paolo Di Lorenzo, Stefania Sardellitti, and Sergio Barbarossa, "Simplicial attention neural networks," *CoRR*, 2022.
- [16] Behrooz Tahmasebi, Derek Lim, and Stefanie Jegelka, "Counting substructures with higher-order graph neural networks: Possibility and impossibility results," *arXiv preprint arXiv:2012.03174*, 2020.
- [17] Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec, "Identity-aware graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 10737–10745.
- [18] Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein, "Improving graph neural network expressivity via subgraph isomorphism counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [19] Ralph Abboud, Ismail Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz, "The surprising power of graph neural networks with random node initialization," in *IJCAI*, 2021.
- [20] 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)*. SIAM, 2021, pp. 333–341.
- [21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, "Computational capabilities of graph neural networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 81–102, 2008.
- [22] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković, "Principal neighbourhood aggregation for graph nets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13260–13271, 2020.
- [23] Dominique Beaini, Saro Passaro, Vincent Létourneau, Will Hamilton, Gabriele Corso, and Pietro Liò, "Directional graph networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 748–758.
- [24] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine, "Breaking the limits of message passing graph neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 599–608.
- [25] Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Survrit Sra, Haggai Maron, and Stefanie Jegelka, "Sign and basis invariant networks for spectral graph representation learning," *arXiv preprint arXiv:2202.13013*, 2022.
- [26] Andreas Loukas, "What graph neural networks cannot learn: depth vs width," in *International Conference on Learning Representations*, 2019.
- [27] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna, "On the equivalence between graph isomorphism testing and function approximation with gnns," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [28] Willem H Haemers and Edward Spence, "Enumeration of cospectral graphs," *European Journal of Combinatorics*, vol. 25, no. 2, pp. 199–211, 2004.
- [29] Aliaksei Sandryhaila and José M.F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [30] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson, "Benchmarking graph neural networks," *arXiv preprint arXiv:2003.00982*, 2020.
- [31] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli, "A fair comparison of graph neural networks for graph classification," *arXiv preprint arXiv:1912.09893*, 2019.