On the equivalence between graph isomorphism testing and function approximation with GNNs

Zhengdao Chen
Courant Institute of Mathematical Sciences
New York University
zc1216@nyu.edu

Soledad Villar
Courant Institute of Mathematical Sciences
Center for Data Science
New York University
soledad.villar@nyu.edu

Lei Chen
Courant Institute of Mathematical Sciences
New York University
lc3909@nyu.edu

Joan Bruna
Courant Institute of Mathematical Sciences
Center for Data Science
New York University
bruna@cims.nyu.edu

Abstract

Graph neural networks (GNNs) have achieved lots of success on graph-structured data. In the light of this, there has been increasing interest in studying their representation power. One line of work focuses on the universal approximation of permutation-invariant functions by certain classes of GNNs, and another demonstrates the limitation of GNNs via graph isomorphism tests.

Our work connects these two perspectives and proves their equivalence. We further develop a framework of the representation power of GNNs with the language of sigma-algebra, which incorporates both viewpoints. Using this framework, we compare the expressive power of different classes of GNNs as well as other methods on graphs. In particular, we prove that order-2 Graph G-invariant networks fail to distinguish non-isomorphic regular graphs with the same degree. We then extend them to a new architecture, Ring-GNNs, which succeeds on distinguishing these graphs as well as for social network datasets.

1 Introduction

Graph structured data naturally occur in many areas of knowledge, including computational biology, chemistry and social sciences. Graph neural networks, in all their forms, yield useful representations of graph data partly because they take into consideration the intrinsic symmetries of graphs, such as invariance and equivariance with respect to a relabeling of the nodes [25, 7, 14, 8, 10, 26, 3].

All these different architectures are proposed with different purposes (see [29] for a survey and references therein), and a priori it is not obvious how to compare their power. The recent work [30] proposes to study the representation power of GNNs via their performance on graph isomorphism tests. They developed the Graph Isomorphism Networks (GINs) that are as powerful as the one-dimensional Weisfeiler-Lehman (1-WL or just WL) test for graph isomorphism [28], and showed that no other neighborhood-aggregating (or message passing) GNN can be more powerful than the 1-WL test. Variants of message passing GNNs include [25, 9].

On the other hand, for feed-forward neural networks, many results have been obtained regarding their ability to approximate continuous functions, commonly known as the universal approximation theorems, such as the seminal works of [6, 12]. Following this line of work, it is natural to study

Preprint. Under review.

the expressivity of graph neural networks in terms of function approximation. Since we could argue that many if not most functions on a graph that we are interested in are invariant or equivariant to permutations of the nodes in the graph, GNNs are usually designed to be invariant or equivariant, and therefore the natural question is whether certain classes GNNs can approximate any continuous and invariant or equivariant functions. Recent work [18] showed the universal approximation of G-invariant networks, constructed based on the linear invariant and equivariant layers studied in [17], if the order of the tensor involved in the networks can grow as the graph gets larger. Such a dependence on the graph size was been theoretically overcame by the very recent work [13], though there is no known upper bound on the order of the tensors involved. With potentially very-high-order tensors, these models that are guaranteed of univeral approximation are not quite feasible in practice.

The foundational part of this work aims at building the bridge between graph isomorphism testing and invariant function approximation, the two main perspectives for studying the expressive power of graph neural networks. We demonstrate an equivalence between the the ability of a class of GNNs to distinguish between any pairs of non-isomorphic graph and its power of approximating any (continuous) invariant functions, for both the case with finite feature space and the case with continuous feature space. Furthermore, we argue that the concept of sigma-algebras on the space of graphs is a natural description of the power of graph neural networks, allowing us to build a taxonomy of GNNs based on how their respective sigmas-algebras interact. Building on this theoretical framework, we identify an opportunity to increase the expressive power of order-2 G-invariant networks with computational tractability, by considering a ring of invariant matrices under addition and multiplication. We show that the resulting model, which we refer to as *Ring-GNN*, is able to distinguish between non-isomorphic regular graphs where order-2 G-invariant networks provably fail. We illustrate these gains numerically in synthetic and real graph classification tasks.

Summary of main contributions:

- We show the equivalence between graph isomorphism testing and approximation of permutation-invariant functions for analyzing the expressive power of graph neural networks.
- We introduce a language of sigma algebra for studying the representation power of graph neural networks, which unifies both graph isomorphism testing and function approximation, and use this framework to compare the power of some GNNs and other methods.
- We propose Ring-GNN, a tractable extension of order-2 Graph G-invariant Networks that uses the ring of matrix addition and multiplication. We show this extension is necessary and sufficient to distinguish Circular Skip Links graphs.

2 Related work

Graph Neural Networks and graph isomorphism. Graph isomorphism is a fundamental problem in theoretical computer science. It amounts to deciding, given two graphs A, B, whether there exists a permutation π such that $\pi A = B\pi$. There exists no known polynomial-time algorithm to solve it, but recently Babai made a breakthrough by showing that it can be solved in quasi-polynomial-time [1]. Recently [30] introduced graph isomorphism tests as a characterization of the power of graph neural networks. They show that if a GNN follows a neighborhood aggregation scheme, then it cannot distinguish pairs of non-isomorphic graphs that the 1-WL test fails to distinguish. Therefore this class of GNNs is at most as powerful as the 1-WL test. They further propose the Graph Isomorphism Networks (GINs) based on approximating injective set functions by multi-layer perceptrons (MLPs), which can be as powerful as the 1-WL test. Based on k-WL tests [4], [19] proposes k-GNN, which can take higher-order interactions among nodes into account. Concurrently to this work, [16] proves that order-k invariant graph networks are at least as powerful as the k-WL tests, and similarly to us, it and augments order-2 networks with matrix multiplication. They show they achieve at least the power of 3-WL test. [20] proposes relational pooling (RP), an approach that combines *permutation-sensitive* functions under all permutations to obtain a permutation-invariant function. If RP is combined with permutation-sensitive functions that are sufficiently expressive, then it can be shown to be a universal approximator. A combination of RP and GINs is able to distinguish certain non-isomorphic regular graphs which GIN alone would fail on. A drawback of RP is that its full version is intractable computationally, and therefore it needs to be approximated by averaging over randomly sampled permutations, in which case the resulting functions is not guaranteed to be permutation-invariant.

Universal approximation of functions with symmetry. Many works have discussed the function approximation capabilities of neural networks that satisfy certain symmetries. [2] studies the symmetry in neural networks from the perspective of probabilistic symmetry and characterizes the deterministic and stochastic neural networks that satisfy certain symmetry. [24] shows that equivariance of a neural network corresponds to symmetries in its parameter-sharing scheme. [31] proposes a neural network architecture with polynomial layers that is able to achieve universal approximation of invariant or equivariant functions. [17] studies the spaces of all invariant and equivariant linear functions, and obtained bases for such spaces. Building upon this work, [18] proposes the G-invariant network for a symmetry group G, which achieves universal approximation of G-invariant functions if the maximal tensor order involved in the network to grow as $\frac{n(n-1)}{2}$, but such high-order tensors are prohibitive in practice. Upper bounds on the approximation power of the G-invariant networks when the tensor order is limited remains open except for when $G = A_n$ [18]. The very recent work [13] extends the result to the equivariant case, although it suffers from the same problem of possibly requiring high-order tensors. Within the computer vision literature, this problem has also been addressed, in particular [11] proposes an architecture that can potentially express all equivariant functions.

To the best our knowledge, this is the first work that shows an explicit connection between the two aforementioned perspectives of studying the representation power of graph neural networks - graph isomorphism testing and universal approximation. Our main theoretical contribution lies in showing an equivalence between them, for both finite and continuous feature space cases, with a natural generalization of the notion of graph isomorphism testing to the latter case. Then we focus on the Graph G-invariant network based on [17, 18], and showed that when the maximum tensor order is restricted to be 2, then it cannot distinguish between non-isomorphic regular graphs with equal degrees. As a corollary, such networks are not universal. Note that our result shows an upper bound on order 2 G-invariant networks, whereas concurrently to us, [16] provides a lower bound by relating to k-WL tests. Concurrently to [16], we propose a modified version of order-2 graph networks to capture higher-order interactions among nodes without computing tensors of higher-order.

3 Graph isomorphism testing and universal approximation

In this section we show that there exists a very close connection between the universal approximation of permutation-invariant functions by a class of functions, and its ability to perform graph isomorphism tests. We consider graphs with nodes and edges labeled by elements of a compact set $\mathcal{X} \subset \mathbb{R}$. We represent graphs with n nodes by an n by n matrix $G \in \mathcal{X}^{n\times n}$, where a diagonal term G_{ii} represents the label of the ith node, and a non-diagonal G_{ij} represents the label of the edge from the ith node to the jth node. An undirected graph will then be represented by a symmetric G.

Thus, we focus on analyzing a collection $\mathcal{C}$ of functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$. We are especially interested in collections of *permutation-invariant functions*, defined so that $f(\pi^\intercal G\pi) = f(G)$, for all $G \in \mathcal{X}^{n\times n}$, and all $\pi \in S_n$, where S_n is the permutation group of n elements. For classes of functions, we define the property of being able to discriminate non-isomorphic graphs, which we call *GIso-discriminating*, which as we will see generalizes naturally to the continuous case.

Definition 1. *Let $\mathcal{C}$ be a collection of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$. We say $\mathcal{C}$ is* ***GIso-discriminating*** *if for all non-isomorphic $G_1, G_2 \in \mathcal{X}^{n\times n}$ (denoted $G_1 \not\simeq G_2$), there exists a function $h \in \mathcal{C}$ such that $h(G_1) \neq h(G_2)$. This definition is illustrated by figure 2 in the appendix.*

Definition 2. *Let $\mathcal{C}$ be a collection of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$. We say $\mathcal{C}$ is* ***universally approximating*** *if for all permutation-invariant function f from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$, and for all $\epsilon > 0$, there exists $h_{f,\epsilon} \in \mathcal{C}$ such that $\|f - h_{f,\epsilon}\|_\infty := \sup_{G\in\mathcal{X}^{n\times n}} |f(G) - h(G)| < \epsilon$*

3.1 Finite feature space

As a warm-up we first consider the space of graphs with a finite set of possible features for nodes and edges, $\mathcal{X} = \{1, \dots, M\}$.

Theorem 1. *Universally approximating classes of functions are also GIso-discriminating.*

Proof. Given $G_1, G_2 \in \mathcal{X}^{n\times n}$, we consider the permutation-invariant function $\mathbb{1}_{\simeq G_1} : \mathcal{X}^{n\times n} \to \mathbb{R}$ such that $\mathbb{1}_{\simeq G_1}(G) = 1$ if G is isomorphic to G_1 and 0 otherwise. Therefore, it can be approximated

with $\epsilon = 0.1$ by a function $h \in \mathcal{C}$. Then h is a function that distinguishes G_1 from G_2, as in Definition 1. Hence $\mathcal{C}$ is GIso-discriminating. □

To obtain a result on the reverse direction, we first introduce the concept of an augmented collection of functions, which is especially natural when $\mathcal{C}$ is a collection of neural networks.

Definition 3. *Given $\mathcal{C}$, a collection of functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$, we consider an augmented collection of functions also from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ consisting of functions that map an input graph G to $\mathcal{NN}([h_1(G), ..., h_d(G)])$ for some finite d, where $\mathcal{NN}$ is a feed-forward neural network / multi-layer perceptron, and $h_1, ..., h_d \in \mathcal{C}$. When $\mathcal{NN}$ is restricted to have L layers, we denoted this augmented collection by $\mathcal{C}^{+L}$. In this work, we consider ReLU as the nonlinear activation function in the neural networks.*

Remark 1. *If $\mathcal{C}_{L_0}$ is the collection of feed-forward neural networks with L_0 layers, then $\mathcal{C}_{L_0}^{+L}$ represents the collection of feed-forward neural networks with $L_0 + L$ layers.*

Remark 2. *If $\mathcal{C}$ is a collection of permutation-invariant functions, so is $\mathcal{C}^{+L}$.*

Theorem 2. *If $\mathcal{C}$ is GIso-discriminating, then $\mathcal{C}^{+2}$ is universal approximating.*

The proof is simple and it is a consequence of the following lemmas that we prove in Appendix A.

Lemma 1. *If $\mathcal{C}$ is GIso-discriminating, then for all $G \in \mathcal{X}^{n\times n}$, there exists a function $\tilde{h}_G \in \mathcal{C}^{+1}$ such that for all G', $\tilde{h}_G(G') = 0$ if and only if $G \simeq G'$.*

Lemma 2. *Let $\mathcal{C}$ be a class of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ satisfying the consequences of Lemma 1, then $\mathcal{C}^{+1}$ is universally approximating.*

3.2 Extension to the case of continuous (Euclidean) feature space

Graph isomorphism is an inherently discrete problem, whereas universal approximation is usually more interesting when the input space is continuous. With our definition 1 of *GIso-discriminating*, we can achieve a natural generalization of the above results to the scenarios of continuous input space. All proofs for this section can be found in Appendix A.

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}$, and we consider graphs with n nodes represented by $G \in K = \mathcal{X}^{n\times n}$; that is, the node features are $\{G_{ii}\}_{i=1,...,n}$ and the edge features are $\{G_{ij}\}_{i,j=1,...,n;i\neq j}$.

Theorem 3. *If $\mathcal{C}$ is universally approximating, then it is also GIso-discriminating*

The essence of the proof is similar to that of Theorem 1. The other direction - showing that pairwise discrimination can lead to universal approximation - is less straightforward. As an intermediate step between, we make the following definition:

Definition 4. *Let $\mathcal{C}$ be a class of functions $K \to \mathbb{R}$. We say it is able to **locate every isomorphism class** if for all $G \in K$ and for all $\epsilon > 0$ there exists $h_G \in \mathcal{C}$ such that:*

- *for all $G' \in K$, $h_G(G') \geq 0$;*
- *for all $G' \in K$, if $G' \simeq G$, then $h_G(G') = 0$; and*
- *there exists $\delta_G > 0$ such that if $h_G < \delta_G$, then $\exists \pi \in S_n$ such that $d(\pi(G'), G) < \epsilon$, where d is the Euclidean distance defined on $\mathbb{R}^{n\times n}$*

Lemma 3. *If $\mathcal{C}$, a collection of continuous permutation-invariant functions from K to $\mathbb{R}$, is GIso-discriminating, then $\mathcal{C}^{+1}$ is able to locate every isomorphism class.*

Heuristically, we can think of the h_G in the definition above as a "loss function" that penalizes the deviation of G' from the equivalence class of G. In particular, the second condition says that if the loss value is small enough, then we know that G' has to be close to the equivalence class of G.

Lemma 4. *Let $\mathcal{C}$ be a class of permutation-invariant functions $K \to \mathbb{R}$. If $\mathcal{C}$ is able to locate every isomorphism class, then $\mathcal{C}^{+2}$ is universally approximating.*

Combining the two lemmas above, we arrive at the following theorem:

Theorem 4. *If $\mathcal{C}$, a collection of continuous permutation-invariant functions from K to $\mathbb{R}$, is GIso-discriminating, then $\mathcal{C}^{+3}$ is universaly approximating.*

4 A framework of representation power based on sigma-algebra

4.1 Introducing sigma-algebra to this context

Let $K = \mathcal{X}^{n\times n}$ be a finite input space. Let $Q_K := K/_{\simeq}$ be the set of isomorphism classes under the equivalence relation of graph isomorphism. That is, for all $\tau \in Q_K, \tau = \{\pi^\intercal G\pi : \pi \in \Gamma_n\}$ for some $G \in K$.

Intuitively, a maximally expressive collection of permutation-invariant functions, $\mathcal{C}$, will allow us to know exactly which isomorphism class τ a given graph G belongs to, by looking at the outputs of certain functions in the collection applied to G. Heuristically, we can consider each function in $\mathcal{C}$ as a "measurement", which partitions that graph space K according to the function value at each point. If $\mathcal{C}$ is powerful enough, then as a collection it will partition K to be as fine as Q_K. If not, it is going to be coarser than Q_K. These intuitions motivate us to introduce the language of sigma-algebra.

Recall that an algebra on a set K is a collection of subsets of K that includes K itself, is closed under complement, and is closed under finite union. Because K is finite, we have that an algebra on K is also a sigma-algebra on K, where a sigma-algebra further satisfies the condition of being closed under countable unions. Since Q_K is a set of (non-intersecting) subsets of K, we can obtain the algebra generated by Q_K, defined as the smallest algebra that contains Q_K, and use $\sigma(Q_K)$ to denote the algebra (and sigma-algebra) generated by Q_K.

Observation 1. *If $f : \mathcal{X}^{n\times n} \to \mathbb{R}$ is a permutation-invariant function, then f is measurable with respect to $\sigma(Q_K)$, and we denote this by $f \in \mathcal{M}[\sigma(Q_K)]$*

Now consider a class of functions $\mathcal{C}$ that is permutation-invariant. Then for all $f \in \mathcal{C}$, $f \in \mathcal{M}[\sigma(Q_K)]$. We define the sigma-algebra generated by f as the set of all the pre-images of Borel sets on $\mathbb{R}$ under f, and denote it by $\sigma(f)$. It is the smallest sigma-algebra on K that makes f measurable. For a class of functions $\mathcal{C}$, $\sigma(\mathcal{C})$ is defined as the smallest sigma-algebra on K that makes all functions in $\mathcal{C}$ measurable. Because here we assume K is finite, it does not matter whether $\mathcal{C}$ is a countable collection.

4.2 Reformulating graph isomorphism testing and universal approximation with sigma-algebra

We restrict our attention to the case of finite feature space. Given a graph $G \in \mathcal{X}^{n\times n}$, we use $\mathcal{E}(G)$ to denote its isomorphism class, $\{G' \in \mathcal{X}^{n\times n} : G' \simeq G\}$. The following results are proven in Section B

Theorem 5. *If $\mathcal{C}$ is a class of permutation-invariant functions on $\mathcal{X}^{n\times n}$ and $\mathcal{C}$ is GIso-discriminating, then $\sigma(\mathcal{C}) = \sigma(Q_K)$*

Together with Theorem 1, the following is an immediate consequence:

Corollary 1. *If $\mathcal{C}$ is a class of permutation-invariant functions on $\mathcal{X}^{n\times n}$ and $\mathcal{C}$ achieves universal approximation, then $\sigma(\mathcal{C}) = \sigma(Q_K)$.*

Theorem 6. *Let be $\mathcal{C}$ a class of permutation-invariant functions on $\mathcal{X}^{n\times n}$ with $\sigma(\mathcal{C}) = \sigma(Q_K)$. Then $\mathcal{C}$ is GIso-discriminating.*

Thus, this sigma-algebra language is a natural notion for characterizing the power of graph neural networks, because as shown above, generating the finest sigma-algebra $\sigma(Q_K)$ is equivalent to being GIso-discriminating, and therefore to universal approximation.

Moreover, when $\mathcal{C}$ is not GIso-discriminating or universal, we can evaluate its representation power by studying $\sigma(\mathcal{C})$, which gives a measure for comparing the power of different GNN families. Given two classes of functions $\mathcal{C}_1, \mathcal{C}_2$, there is $\sigma(\mathcal{C}_1) \subseteq \sigma(\mathcal{C}_2)$ if and only if $\mathcal{M}[\sigma(\mathcal{C}_1)] \subseteq \mathcal{M}[\sigma(\mathcal{C}_2)]$ if and only if $\mathcal{C}_1$ is less powerful than $\mathcal{C}_2$ in terms of representation power.

In Appendix C we use this notion to compare the expressive power of different families of GNNs as well as other algorithms like 1-WL, linear programming and semidefinite programming in terms of their ability to distinguish non-isomorphic graphs. We summarize our findings in Figure 1.

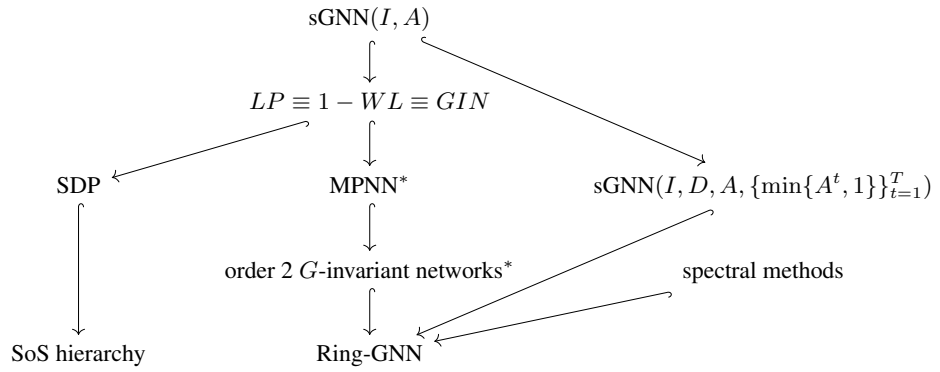

Figure 1: Relative comparison of function classes in terms of their ability to solve graph isomorphism.
*Note that, on one hand GIN is defined by [30] as a form of message passing neural network justifying the inclusion GIN $\hookrightarrow$ MPNN. On the other hand [17] shows that message passing neural networks can be expressed as a modified form of order 2 G-invariant networks (which may not coincide with the definition we consider in this paper). Therefore the inclusion GIN $\hookrightarrow$ order 2 G-invariant networks has yet to be established rigorously.

5 Ring-GNN: a GNN defined on the ring of equivariant functions

We now investigate the G-invariant network framework proposed in [18] (see Appendix D for its definition and a description of an adapted version that works on graph-structured inputs, which we call the *Graph G-invariant Networks*). The architecture of G-invariant networks is built by interleaving compositions of equivariant linear layers between tensors of potentially different orders and point-wise nonlinear activation functions. It is a powerful framework that can achieve universal approximation if the order of the tensor can grow as $\frac{n(n-1)}{2}$, where n is the number of nodes in the graph, but less is known about its approximation power when the tensor order is restricted. One particularly interesting subclass of G-invariant networks is the ones with maximum tensor order 2, because [17] shows that it can approximate any Message Passing Neural Network [8]. Moreover, it is both mathematically cumbersome and computationally expensive to include equivariant linear layers involving tensors with order higher than 2.

Our following result shows that the order-2 Graph G-invariant Networks subclass of functions is quite restrictive. The proof is given in Appendix D.

Theorem 7. *Order-2 Graph G-invariant Networks cannot distinguish between non-isomorphic regular graphs with the same degree.*

Motivated by this limitation, we propose a GNN architecture that extends the family of order-2 Graph G-invariant Networks without going into higher order tensors. In particular, we want the new family to include GNNs that can distinguish some pairs of non-isomorphic regular graphs with the same degree. For instance, take the pair of Circular Skip Link graphs $G_{8,2}$ and $G_{8,3}$, illustrated in Figure 5. Roughly speaking, if all the nodes in both graphs have the same node feature, then because they all have the same degree, the updates of node states in both graph neural networks based on neighborhood aggregation and the WL test will fail to distinguish the nodes. However, the *power graphs*[1] of $G_{8,2}$ and $G_{8,3}$ have different degrees. Another important example comes from spectral methods that operate on *normalized* operators, such as the normalized Laplacian $\Delta = I - D^{-1/2}AD^{-1/2}$, where D is the diagonal degree operator. Such normalization preserves the permutation symmetries and in many clustering applications leads to dramatic improvements [27].

This motivates us to consider a polynomial ring generated by the matrices that are the outputs of permutation-equivariant linear layers, rather than just the linear space of those outputs. Together with point-wise nonlinear activation functions such as ReLU, power graph adjacency matrices like $\min(A^2, 1)$ can be expressed with suitable choices of parameters. We call the resulting architecture the *Ring-GNN* [2].

[1] If A is the adjacency matrix of a graph, its power graph has adjacency matrix $\min(A^2, 1)$. The matrix $\min(A^2, 1)$ has been used in [5] in graph neural networks for community detection and in [21] for the quadratic assignment problem.

[2] We call it Ring-GNN since the main object we consider is the ring of matrices, but technically we can express an associative algebra since our model includes scalar multiplications.

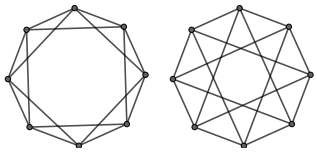

Figure 2: The Circular Skip Link graphs $G_{n,k}$ are undirected graphs in n nodes $q_0, \ldots, q_{n-1}$ so that $(i,j) \in E$ if and only if $|i-j| \equiv 1$ or $k \pmod n$. In this figure we depict (left) $G_{8,2}$ and (right) $G_{8,3}$. It is very easy to check that $G_{n,k}$ and $G_{n',k'}$ are not isomorphic unless $n = n'$ and $k \equiv \pm k' \pmod n$. Both 1-WL and G-invariant networks fail to distinguish them.

Definition 5 (Ring-GNN)**.** *Given a graph in n nodes with both node and edge features in $\mathbb{R}^d$, we represent it with a matrix $A \in \mathbb{R}^{n\times n\times d}$. [17] shows that all linear equivariant layers from $\mathbb{R}^{n\times n}$ to $\mathbb{R}^{n\times n}$ can be expressed as $L_\theta(A) = \sum_{i=1}^{15} \theta_i L_i(A) + \sum_{i=16}^{17} \theta_i \overline{L}_i$, where the $\{L_i\}_{i=1,\ldots,15}$ are the 15 basis functions of all linear equivariant functions from $\mathbb{R}^{n\times n}$ to $\mathbb{R}^{n\times n}$, $\overline{L}_{16}$ and $\overline{L}_{17}$ are the basis for the bias terms, and $\theta \in \mathbb{R}^{17}$ are the parameters that determine L. Generalizing to an equivariant linear layer from $\mathbb{R}^{n\times n\times d}$ to $\mathbb{R}^{n\times n\times d'}$, we set $L_\theta(A)_{\cdot,\cdot,k'} = \sum_{k=1}^{d} \sum_{i=1}^{15} \theta_{k,k',i} L_i(A_{\cdot,\cdot,i}) + \sum_{i=16}^{17} \theta_{k,k',i} \overline{L}_i$, with $\theta \in \mathbb{R}^{d\times d'\times 17}$.*

With this formulation, we now define a Ring-GNN with T layers. First, set $A^{(0)} = A$. In the t^{th} layer, let

$$\begin{aligned} B_1^{(t)} &= \rho(L_{\alpha^{(t)}}(A^{(t)})) \\ B_2^{(t)} &= \rho(L_{\beta^{(t)}}(A^{(t)}) \cdot L_{\gamma^{(t)}}(A^{(t)})) \\ A^{(t+1)} &= k_1^{(t)} B_1^{(t)} + k_2^{(t)} B_2^{(t)} \end{aligned}$$

where $k_1^{(t)}, k_2^{(t)} \in \mathbb{R}$, $\alpha^{(t)}, \beta^{(t)}, \gamma^{(t)} \in \mathbb{R}^{d^{(t)}\times d'^{(t)}\times 17}$ are learnable parameters. If a scalar output is desired, then in the general form, we set the output to be $\theta_S \sum_{i,j} A_{ij}^{(T)} + \theta_D \sum_{i,i} A_{ii}^{(T)} + \sum_i \theta_i \lambda_i(A^{(T)})$, where $\theta_S, \theta_D, \theta_1, \ldots, \theta_n \in \mathbb{R}$ are trainable parameters, and $\lambda_i(A^{(T)})$ is the i-th eigenvalue of $A^{(L)}$.

Note that each layer is equivariant, and the map from A to the final scalar output is invariant. A Ring-GNN can reduce to an order-2 Graph G-invariant Network if $k_2^{(t)} = 0$. With $J+1$ layers and suitable choices of the parameters, it is possible to obtain $\min(A^{2^J}, 1)$ in the $(J+1)^{th}$ layer. Therefore, we expect it to succeed in distinguishing certain pairs of regular graphs that order-2 Graph G-invariant Networks fail on, such as the Circular Skip Link graphs. Indeed, this is verified in the synthetic experiment presented in the next section. The normalized Laplacian can also be obtained, since the degree matrix can be inverted by taking the reciprocal on the diagonal, and then entry-wise inversion and square root on the diagonal can be approximated by MLPs.

The terms in the output layer involving eigenvalues are optional, depending on the task. For example, in community detection spectral information is commonly used [15]. We could also take a fixed number of eigenvalues instead of the full spectrum. In the experiments, Ring-GNN-SVD includes the eigenvalue terms while Ring-GNN does not, as explained in appendix E. Computationally, the complexity of running the forward model grows as $O(n^3)$, dominated by matrix multiplications and possibly singular value decomposition for computing the eigenvalues. We note also that Ring-GNN can be augmented with matrix inverses or more generally with functional calculus on the spectrum of any of the intermediate representations [3] while keeping $O(n^3)$ computational complexity. Finally, note that a Graph G-invariant Network with maximal tensor order d will have complexity at least $O(n^d)$. Therefore, the Ring-GNN explores higher-order interactions in the graph that order-2 Graph G-invariant Networks neglects while remaining computationally tractable.

6 Experiments

The different models and the detailed setup of the experiments are discussed in Appendix E.

[3] When $A = A^{(0)}$ is an undirected graph, one easily verifies that $A^{(t)}$ contains only symmetric matrices for each t.

6.1 Classifying Circular Skip Links (CSL) graphs

The following experiment on synthetic data demonstrates the connection between function fitting and graph isomorphism testing. The Circular Skip Links graphs are undirected regular graphs with node degree 4 [20], as illustrated in Figure 5. Note that two CSL graphs $G_{n,k}$ and $G_{n',k'}$ are not isomorphic unless $n = n'$ and $k \equiv \pm k' \pmod{n}$. In the experiment, which has the same setup as in [20], we fix $n = 41$, and set $k \in \{2, 3, 4, 5, 6, 9, 11, 12, 13, 16\}$, and each k corresponds to a distinct isomorphism class. The task is then to classify a graph $G_{n,k}$ by its skip length k.

Note that since the 10 classes have the same size, a naive uniform classifier would obtain 0.1 accuracy. As we see from Table 1, both GIN and G-invariant network with tensor order 2 do not outperform the naive classifier. Their failure in this task is unsurprising: WL tests are proved to fall short of distinguishing such pairs of non-isomorphic regular graphs [4], and hence neither can GIN [30]; by the theoretical results from the previous section, order-2 Graph G-invariant network are unable to distinguish them either. Therefore, their failure as graph isomorphism tests is consistent with their failure in this classification task, which can be understood as trying to approximate the function that maps the graph to their class labels.

It should be noted that, since graph isomorphism tests are not entirely well-posed as classfication tasks, the performance of GNN models could vary due to randomness. But the fact that Ring-GNNs achieve a relatively high maximum accuracy (compared to RP for example) demonstrates that as a class of GNNs it is rich enough to contain functions that distinguish the CSL graphs to a large extent.

	Circular Skip Links			IMDBB		IMDBM	
GNN architecture	max	min	std	mean	std	mean	std
RP-GIN †	53.3	10	12.9	-	-	-	-
GIN †	10	10	0	75.1	5.1	52.3	2.8
Order 2 G-invariant †	10	10	0	71.27	4.5	48.55	3.9
sGNN-5	80	80	0	72.8	3.8	49.4	3.2
sGNN-2	30	30	0	73.1	5.2	49.0	2.1
sGNN-1	10	10	0	72.7	4.9	49.0	2.1
LGNN [5]	30	30	0	74.1	4.6	50.9	3.0
Ring-GNN	80	10	15.7	73.0	5.4	48.2	2.7
Ring-GNN-SVD	100	100	0	73.1	3.3	49.6	3.0

Table 1: **(left)** Accuracy of different GNNs at classifying CSL (see Section 6.1). We report the best performance and worst performance among 10 experiments. **(right)** Accuracy of different GNNs at classifying real datasets (see Section 6.1). We report the best performance among all epochs on a 10-fold cross validation dataset, as was done in [30]. †: Reported performance by [20], [30] and [17].

6.2 IMDB datasets

We use the two IMDB datasets (IMDBBINARY, IMDBMULTI) to test different models in real-world scenarios. Since our focus is on distinguishing graph structures, these datasets are suitable as they do not contain node features, and hence the adjacency matrix contains all the input data. IMDBBINARY dataset has 1000 graphs, with average number of nodes 19.8 and 2 classes. The dataset is randomly partitioned into 900/100 for training/validation. IMDBMULTI dataset has 1500 graphs, with average number of nodes 13.0 and 3 classes. The dataset is randomly partitioned into 1350/150 for training/validation. All models are evaluated via 10-fold cross validation and best accuracy is calculated through averaging across folds followed by maximizing along epochs [30]. Importantly, the architecture hyper-parameter of Ring-GNN we use is close to that provided in [17] to show that order-2 G-invariant Network is included in model family we propose. The results show that Ring-GNN models achieve higher performance than Order-2 G-invariant networks in both datasets. Admittedly its accuracy does not reach that of the state-of-the-art. However, the main goal of this part of our work is not necessarily to invent the best-performing GNN through hyperparameter optimization, but rather to propose Ring-GNN as an augmented version of order-2 Graph G-invariant Networks and show experimental results that support the theory.

7 Conclusions

In this work we address the important question of organizing the fast-growing zoo of GNN architectures in terms of what functions they can and cannot represent. We follow the approach via the graph isomorphism test, and show that is equivalent to the other perspective via function approximation. We leverage our graph isomorphism reduction to augment order-2 G-invariant nets with the ring of operators associated with matrix multiplication, which gives provable gains in expressive power with complexity $O(n^3)$, and is amenable to efficiency gains by leveraging sparsity in the graphs.

Our general framework leaves many interesting questions unresolved. First, a more comprehensive analysis on which elements of the algebra are really needed depending on the application. Next, our current GNN taxonomy is still incomplete, and in particular we believe it is important to further discern the abilities between spectral and neighborhood-aggregation-based architectures. Finally, and most importantly, our current notion of invariance (based on permutation symmetry) defines a topology in the space of graphs that is too strong; in other words, two graphs are either considered equal (if they are isomorphic) or not. Extending the theory of symmetric universal approximation to take into account a weaker metric in the space of graphs, such as the Gromov-Hausdorff distance, is a natural next step, that will better reflect the stability requirements of powerful graph representations to small graph perturbations in real-world applications.

Acknowledgements We would like to thank Haggai Maron for fruitful discussions and for pointing us towards G-invariant networks as powerful models to study representational power in graphs. This work was partially supported by NSF grant RI-IIS 1816753, NSF CAREER CIF 1845360, the Alfred P. Sloan Fellowship, Samsung GRP and Samsung Electronics. SV was partially funded by EOARD FA9550-18-1-7007 and the Simons Collaboration Algorithms and Geometry.

References

[1] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697. ACM, 2016.

[2] Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetry and invariant neural networks. *arXiv preprint arXiv:1901.06082*, 2019.

[3] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017.

[4] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[5] Zhengdao Chen, Lisha Li, and Joan Bruna. Supervised community detection with line graph neural networks. *Internation Conference on Learning Representations*, 2019.

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

[7] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[8] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[10] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[11] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems*, pages 7211–7221, 2018.

[12] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[13] Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *arXiv preprint arXiv:1905.04943*, 2019.

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

[15] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[16] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Lipman Yaron. Provably powerful graph networks. *arXiv preprint arXiv:1905.11136*, 2019.

[17] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. 2018.

[18] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. *arXiv preprint arXiv:1901.09342*, 2019.

[19] 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. *Association for the Advancement of Artificial Intelligence*, 2019.

[20] Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. *arXiv preprint arXiv:1903.02541*, 2019.

[21] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. A note on learning algorithms for quadratic assignment with graph neural networks. *arXiv preprint arXiv:1706.07450*, 2017.

[22] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, lasserre gaps, and asymmetry of random graphs. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1659–1677. Society for Industrial and Applied Mathematics, 2014.

[23] Motakuri V Ramana, Edward R Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1-3):247–265, 1994.

[24] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[25] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

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

[27] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[28] B Weisfeiler and A Leman. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia*, 2(9):12-16, 1968.

[29] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[30] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[31] Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *arXiv preprint arXiv:1804.10306*, 2018.

[32] Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.

A Proofs on universal approximation and graph isomorphism

Lemma 1. If $\mathcal{C}$ is GIso-discriminating, then for all $G \in \mathcal{X}^{n\times n}$, there exists a function $\tilde{h}_G \in \mathcal{C}^{+1}$ such that for all $G', \tilde{h}_G(G') = 0$ if and only if $G \simeq G'$.

Proof of Lemma 1. Given $G, G' \in \mathcal{X}^{n\times n}$ with $G \not\simeq G'$, let $h_{G,G'} \in \mathcal{C}$ be the function that distinguishes this pair, i.e. $h_{G,G'}(G) \neq h_{G,G'}(G')$. Then define a function $\overline{h}_{G,G'}$ by

$$\begin{aligned}\overline{h}_{G,G'}(G^*) &= |h_{G,G'}(G^*) - h_{G,G'}(G)| \\ &= \max(h_{G,G'}(G^*) - h_{G,G'}(G), 0) + \max(h_{G,G'}(G) - h_{G,G'}(G^*), 0)\end{aligned} \tag{1}$$

Note that if $G^* \simeq G$, then $h_{G,G'}(G^*) = h_{G,G'}(G)$, and so $\overline{h}_{G,G'}(G^*) = 0$. If $G^* \simeq G'$, then $\overline{h}_{G,G'}(G^*) > 0$. Otherwise, $\overline{h}_{G,G'}(G^*) \geq 0$.

Next, define a function $\tilde{h}_G$ by $\tilde{h}_G(G^*) = \sum_{G'\in\mathcal{X}^{n\times n}, G'\not\simeq G} \overline{h}_{G,G'}(G^*)$. If $G^* \simeq G$, we have $\tilde{h}_G(G^*) = 0$, whereas if $G^* \not\simeq G$ then $\tilde{h}_G(G^*) > 0$.

Thus, it suffices to show that $\tilde{h}_G \in \mathcal{C}^{+1}$. We take the finite subcollection of functions, $\{h_{G,G'}\}_{G'\in\mathcal{X}^{n\times n}, G\not\simeq G'}$, and feed the input graph G' to each of them to obtain a vector of outputs. By equation 1, $\overline{h}_{G,G'}(G^*)$ can be obtained from $h_{G,G'}(G^*)$ by passing through one ReLU layer. Finally, a finite summation across $G' \not\simeq G$ yields $\tilde{h}_G(G^*)$. Therefore, $\tilde{h}_G \in \mathcal{C}^{+1}, \forall G \in \mathcal{X}^{n\times n}$. □

Lemma 2 Let $\mathcal{C}$ be a class of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ so that for all $G \in \mathcal{X}^{n\times n}$, there exists $\tilde{h}_G \in \mathcal{C}$ satisfying $\tilde{h}_G(G') = 0$ if and only if $G \simeq G'$. Then $\mathcal{C}^{+1}$ is universally approximating.

Proof of Lemma 2. In fact, in the finite feature setting we can obtain a stronger result: for all f that is permutation-invariant, $f \in \mathcal{C}^{+1}$, and so no approximation is needed.

We first use the $\tilde{h}_G$'s to construct all the indicator functions $\mathbb{1}_{G\simeq G^*}$ as functions of G^* on $\mathcal{X}^{n\times n}$. To achieve this, because $\mathcal{X}^{n\times n}$ is finite, $\forall G$, we let $\delta_G = \frac{1}{2}\min_{G'\in\mathcal{X}^{n\times n}, G'\not\simeq G} |\tilde{h}_G(G')| > 0$. We then introduce a "bump" function from $\mathbb{R}$ to $\mathbb{R}$ with parameters a and b, $\psi_{a,b}(x) = \psi((x-b)/a)$, where $\psi(x) = \max(x-1, 0) + \max(x+1, 0) - 2\max(x, 0)$. Then $\psi_{a,b}(b) = 0$, and $supp(\psi_{a,b}) = (b - a, b + a)$. Now, we define a function φ_G from $\mathcal{X} = \{1, ..., M\}$ to $\mathbb{R}$ by $\varphi_G(G^*) = \psi_{\delta_G,0}(\tilde{h}_G(G^*))$. Note that $\varphi_G(G^*) = \mathbb{1}_{G\simeq G^*}$ as a function of G^* on $\mathcal{X}^{n\times n}$.

Given f, thanks to the finiteness of the input space $\mathcal{X}^{n\times n}$, we decompose it as $f(G^*) = (\frac{1}{|S_n|}\sum_{G\in\mathcal{X}^{n\times n}} \mathbb{1}_{G\simeq G^*}) f(G^*) = \frac{1}{|S_n|}\sum_{G\in\mathcal{X}^{n\times n}} f(G)\mathbb{1}_{G\simeq G^*} = \frac{1}{|S_n|}\sum_{G\in\mathcal{X}^{n\times n}} f(G)\varphi_G(G^*)$.

The right hand side can be realized in $\mathcal{C}^{+1}$, since we can first take the finite collection of functions $\{\tilde{h}_G\}_{G\in\mathcal{X}^{n\times n}}$ and obtain $\{\tilde{h}_G(G^*)\}_{G\in\mathcal{X}^{n\times n}}$. Then, with an MLP with one hidden layer, we can obtain $\{\varphi_G(G^*)\}_{G\in\mathcal{X}^{n\times n}}$, a linear combination of which gives the right hand side, since each "$f(G)$" within the summation is a constant. □

Theorem 3. If $\mathcal{C}$ is universally approximating, then it is also GIso-discriminating

Proof of Theorem 3. $\forall G_1, G_2 \in K$, if $G_1 \not\simeq G_2$, define $f_1(G) = \min_{\pi\in S_n} d(G_1, \pi^\intercal G\pi)$. It is a continuous and permutation-invariant function on K, and therefore can be approximated by a function $h \in \mathcal{C}$ to within $\epsilon = \frac{1}{2}f_1(G_2) > 0$ accuracy. Then h is a function that can discriminate between G_1 and G_2. □

Lemma 3. If $\mathcal{C}$, a collection of continuous permutation-invariant functions from K to $\mathbb{R}$, is pairwise distinguishing, then $\mathcal{C}^{+1}$ is able to locate every isomorphism class.

Proof of Lemma 3. Fix any $G \in K$. $\forall G' \not\simeq G \in K, \exists h_{G,G'} \in \mathcal{C}$ such that $h_{G,G'}(G) \neq h_{G,G'}(G')$. For each G', define a set $A_{G'}$ as $h^{-1}_{G,G'}((h_{G,G'}(G') - \frac{|h_{G,G'}(G') - h_{G,G'}(G)|}{2}, h_{G,G'}(G') +$

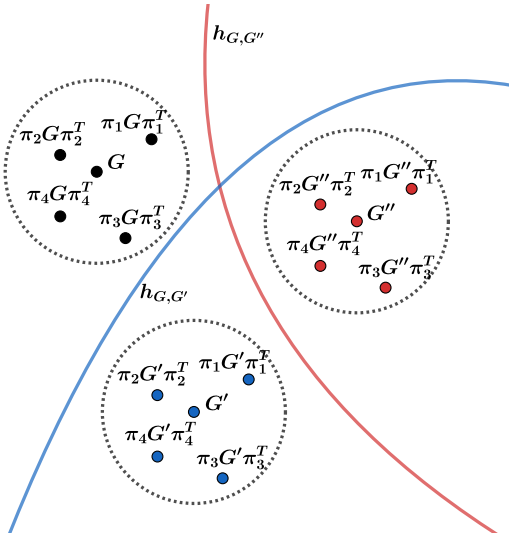

Figure 3: Illustrating the definition of GIso-discriminating. G, G' and G'' are mutually non-isomorphic, and each of the big circles with dashed boundary represents an equivalence class under graph isomorphism. $h_{G,G'}$ is a permutation-invariant function that obtains different values on equivalence class of G and on that of G', and similar $h_{G,G''}$. If the graph space has only these three equivalence classes of graphs, then $\mathcal{C} = \{h_{G,G'}, h_{G,G''}\}$ is GIso-discriminating.

$\frac{|h_{G,G'}(G')-h_{G,G'}(G)|}{2})) \subseteq K$. Obviously $G' \in A_{G'}$ and G does not. Since $h_{G,G'}$ is assumed continuous, A'_G is an open set for each $G' \not\simeq G$. If $G' \simeq G$, define $A_{G'} = B(G', \epsilon)$, the open ϵ-ball in K under the Euclidean distance.

Thus, $\{A_{G'}\}_{G' \in K}$ is an open cover of K. Since K is compact, $\exists$ a finite subset K_0 of K such that $\{A_{G'}\}_{G' \in K_0}$ also covers K.

Hence, $\forall G^* \in K, \exists G' \in K_0$ such that $G^* \in A_{G'}$. Moreover, $\forall G^* \in K \setminus (\bigcup_{G' \in \mathcal{E}(G)} A_{G'}) = K \setminus (\bigcup_{\pi \in S_n} B(\pi^\intercal G \pi, \epsilon))$, where $\mathcal{E}(G)$ represents the equivalence class of graphs in K consisting of graphs isomorphic to G, $\exists G' \in K_0 \setminus \mathcal{E}(G)$ such that $G^* \in A_{G'}$.

Now define a function $\tilde{h}_G$ on K by $\tilde{h}_G(G^*) = \sum_{G' \in K_0 \setminus \mathcal{E}(G)} \overline{h}_{G,G'}(G^*)$, where $\overline{h}_{G,G'}(G^*) = \max(\frac{2}{3}|h_{G,G'}(G) - h_{G,G'}(G')| - |h_{G,G'}(G^*) - h_{G,G'}(G')|, 0)$. Since each $h_{G,G'}$ in continuous, $\tilde{h}_G$ is also continuous. Thus, we can show that $\tilde{h}_G$ is the desired function in Definition 4:

- $h_{G,G'}$ is nonnegative $\forall G, G'$, and hence $\tilde{h}_G$ is nonnegative on K
- If $G^* \simeq G$, then as each $h_{G,G'}$ is permutation invariant, there is $h_{G,G'}(G^*) = h_{G,G'}(G)$, and hence $\overline{h}_{G,G'}(G^*) = 0$. Thus, $\tilde{h}_G(G^*) = 0$.
- If $\forall \pi \in S_n, d(\pi^\intercal G^* \pi, G) \geq \epsilon$, then $G^* \in K \setminus \bigcup_{G' \in \mathcal{E}(G)} A_{G'}$. Therefore, $\exists G' \in K \setminus \mathcal{E}(G)$ such that $G^* \in A_{G'}$, which implies that $|h_{G,G'}(G^*) - h_{G,G'}(G')| < \frac{1}{2}|h_{G,G'}(G) - h_{G,G'}(G')| < \frac{2}{3}|h_{G,G'}(G) - h_{G,G'}(G')|$. Therefore, $\frac{2}{3}|h_{G,G'}(G) - h_{G,G'}(G')| - |h_{G,G'}(G^*) - h_{G,G'}(G')| > \frac{1}{6}|h_{G,G'}(G) - h_{G,G'}(G')| > 0$, and so $\tilde{h}_G(G^*) \geq \overline{h}_{G,G'}(G^*) > \frac{1}{6}|h_{G,G'}(G) - h_{G,G'}(G')|$. Define $\delta_G = \frac{1}{6} \min_{G' \in K_0 \setminus \mathcal{E}(G)} |h_{G,G'}(G) - h_{G,G'}(G')| > 0$. Then if $\tilde{h}_G(G^*) < \delta_G$, it has to be the case that $G^* \in \bigcup_{G' \in \mathcal{E}(G)} A_{G'} = \bigcup_{\pi \in S_n} B(\pi^\intercal G \pi, \epsilon)$, implying that $\exists \pi \in S_n$ such that $d(G^*, \pi^\intercal G \pi) < \epsilon$.

Finally, it is clear that $\tilde{h}_G$ can be realized in $\mathcal{C}^{+1}$.

□

Lemma 4. Let $\mathcal{C}$ be a class of permutation-invariant functions $K \to \mathbb{R}$. If $\mathcal{C}$ is able to locate every isomorphism class, then $\mathcal{C}^{+2}$ is universally approximating.

Proof of Lemma 4. Consider any f that is continuous and permutation-invariant. Since K is compact, f is uniformly continuous on K. Therefore, $\forall \epsilon > 0, \exists r > 0$ such that $\forall G_1, G_2 \in K$, if $d(G_1, G_2) < r$, then $|f(G_1) - f(G_2)| < \epsilon$.

Given $\forall G \in K$, choose the function h_G in definition 2. Use $h_G^{-1}(a)$ to denote $h_G^{-1}([0, a))$. Then $\exists \delta_G$ such that $h_G^{-1}(\delta_G) \subseteq B(G, r)$, where $B(G, r)$ is the ball in K centered at G with radius r (in Euclidean distance). Since h_G is continuous, $h_G^{-1}(\delta_G)$ is open. Therefore, $\{h_G^{-1}(\delta_G)\}_{G \in K}$ is an open cover of K. Because K is compact, $\exists$ a finite subset $K_0 \subseteq K$ such that $\{h_G^{-1}(\delta_G)\}_{G \in K_0}$ also covers K.

$\forall G_0 \in K_0$, define another function $\varphi_{G_0}(G') = \delta_{G_0} - h_{G_0}(G')$ if $h_{G_0}(G') < \delta_{G_0}$ and 0 otherwise. Therefore, $\text{supp}(\varphi_{G_0}) = h_{G_0}^{-1}(\delta_{G_0})$. Let $\varphi(G') = \sum_{G^* \in K_0} \varphi_{G^*}(G')$, and then define $\psi_{G_0}(G') = \frac{\varphi_{G_0}(G')}{\varphi(G')}$. Note that $\forall G' \in K$, since $\{h_G^{-1}(\delta_G)\}_{G \in K_0}$ covers K, $\exists G^* \in K_0$ such that $G' \in h_{G^*}^{-1}(\delta_{G^*}) = \text{supp}(\varphi_{G^*})$, and so the denominator > 0. Therefore, ψ_{G_0} is well defined on K, and $\text{supp}(\psi_{G_0}) = \text{supp}(\varphi_{G_0}) = h_{G_0}^{-1}(\delta_{G_0})$. Moreover, $\forall G' \in K, \sum_{G_0 \in K_0} \psi_{G_0}(G') = 1$. Therefore, the set of functions $\{\psi_{G_0}\}_{G_0 \in K_0}$ is a "partition of unity", with respect to the open cover $\{h_G^{-1}(\delta_G)\}_{G \in K_0}$.

Back to the function f that we want to approximate. We want to express it in away that resembles what a neural network can do. With the set of functions $\{\psi_{G_0}\}_{G_0 \in K_0}$, we have

$$f(G') = \sum_{G_0 \in K_0} f(G')\psi_{G_0}(G') = \sum_{\substack{G_0 \in K_0 \\ G' \in h_{G_0}^{-1}(\delta_{G_0})}} f(G')\psi_{G_0}(G')$$

If $G' \in h_{G_0}^{-1}(\delta_{G_0}$, then $d(G', G_0) > r$, and therefore $|f(G') - f(G_0)| < \epsilon$. Hence, we can use $\bar{h}(G') = \sum_{G_0 \in K_0} f(G_0)\psi_{G_0}(G')$ to approximate $f(G')$, because

$$\begin{aligned} |f(G') - \sum_{G_0 \in K_0} f(G_0)\psi_{G_0}(G')| =& |f(G') - \sum_{\substack{G_0 \in K_0 \\ G' \in h_{G_0}^{-1}(\delta_{G_0})}} f(G_0)\psi_{G_0}(G')| \\ =& \sum_{\substack{G_0 \in K_0 \\ G' \in h_{G_0}^{-1}(\delta_{G_0})}} |f(G') - f(G_0)|\psi_{G_0}(G') \\ <& \epsilon \end{aligned} \tag{2}$$

Finally, we need to show how to approximate $\bar{h}$ with functions from $\mathcal{C}$ augmented with a multi-layer perceptron. We start with $\{h_{G_0}\}_{G_0 \in K} \subseteq \mathcal{C}$, and apply them to the input graph G'. Then, for each of $h_{G_0}G'()$ apply an MLP with one hidden layer to obtain $\varphi_{G_0}(G')$, and use one node to store. their sum, $\varphi(G')$. We then use an MLP with one hidden layer to approximate division, obtaining $\psi_{G_0}(G')$. Finally, $\bar{h}(G')$ is approximated by a linear combination of $\{\psi_{G_0}(G')\}_{G_0 \in K}$, since each $f(G_0)$ is a constant.

□

B Proofs of Section 4.2

Theorem 5. If $\mathcal{C}$ is a class of permutation-invariant functions on $\mathcal{X}^{n \times n}$ and $\mathcal{C}$ is GIso-discriminating, then $\sigma(\mathcal{C}) = \sigma(Q_K)$

Proof of Theorem 5. If $\mathcal{C}$ is GIso-discriminating, then given a $G \in \mathcal{X}^{n \times n}$, $\forall G' \not\simeq G, \exists h_{G'} \in \mathcal{C}$ and $b_{G'} \in \mathbb{R}$ such that $\mathcal{E}(G) = \cap_{G' \not\simeq G} h_{G'}^{-1}(\{b'_G\})$, which is a finite intersection of sets in $\sigma(\mathcal{C})$. Hence, $\mathcal{E}(G) \in \sigma(f_G) \subseteq \sigma(\mathcal{C})$. Therefore, $Q_K \subseteq \sigma(\mathcal{C})$, and hence $\sigma(Q_K) \subseteq \sigma(\mathcal{C})$. Moreover, since $\sigma(g) \subseteq \sigma(Q_K)$ for all $g \in \mathcal{C}$, there is $\sigma(\mathcal{C}) \subseteq \sigma(Q_K)$ □

Theorem 6. Let be $\mathcal{C}$ a class of permutation-invariant functions on $\mathcal{X}^{n\times n}$ with $\sigma(\mathcal{C}) = \sigma(Q_K)$. Then $\mathcal{C}$ is GIso-discriminating.

Proof of Theorem 6. Suppose not. This implies that $Q_K \subsetneq \sigma(\mathcal{C})$, and hence $\exists \tau = \mathcal{E}(G) \in Q_K$ such that $\tau \notin \sigma(\mathcal{C})$. Note that τ is an equivalence class of graphs that are isomorphic to each other. Then consider the smallest subset in $\sigma(\mathcal{C})$ that contains τ, defined as $S(\tau) = \bigcap_{\substack{T\in\sigma(\mathcal{C})\\ \tau\subseteq T}} T$.

Since K is a finite space, $\sigma(\mathcal{C})$ is also finite, and hence this is a finite intersection. Since a sigma-algebra is closed under finite intersection, there is $S(\tau) \in \sigma(C)$. As $\tau \notin \sigma(\mathcal{C})$, we know that $\tau \subsetneq S(\tau)$. Then, $\exists G' \not\simeq G$ such that $G' \in S(\tau)$. Then there does not exist any function h in $\mathcal{C}$ such that $h(G) \neq h(G')$, since otherwise the pre-image of some interval in $\mathbb{R}$ under h will intersect with only $\mathcal{E}(G)$ but not $\mathcal{E}(G')$. Contradiction. □

C Comparison of expressive power of families of functions via graph isomorphism

Given two classes of functions $\mathcal{C}_1, \mathcal{C}_2$, such as two classes of GNNs, there are four possibilities regarding their relative representation power, using the language of sigma-algebra developed in the main text:

- $\sigma(\mathcal{C}_1) = \sigma(\mathcal{C}_2)$
- $\sigma(\mathcal{C}_1) \subsetneq \sigma(\mathcal{C}_2)$
- $\sigma(\mathcal{C}_2) \subsetneq \sigma(\mathcal{C}_1)$
- Not comparable / None of the above (i.e., $\sigma(\mathcal{C}_1) \not\subseteq \sigma(\mathcal{C}_2)$ and $\sigma(\mathcal{C}_1) \not\subseteq \sigma(\mathcal{C}_2)$)

In this section we summarize some results from the literature and show partial relationships between different GNNs architectures in terms of their ability to distinguish non-isomorphic graphs (in the context of the sigma algebra introduced in Section 4). For simplicity, in this section we assume that graphs are given by an adjacency matrix (no node nor edge features are considered). We illustrate our findings in Figure 1.

- **sGNN**$(\mathcal{M})$. We consider spectral GNNs as the ones used in [5] for community detection. In this context we focus on the simplified version where the GNNs are defined as

$$\begin{aligned} v^0 =& \mathbb{1}_n \\ v^{t+1} =& \rho\left(\sum_{M\in\mathcal{M}} M v^t \theta_M^t\right) \text{ where } \theta_M^t \in \mathbb{R}^{d_t\times d_{t+1}} \text{ learnable parameters, } v^t \in \mathbb{R}^{n\times d_t} \\ \text{output} :& \sum_{i=1}^{d_L} v_i^L. \end{aligned}$$

  Usually $\mathcal{M}$ is a set of operators related to the graph. In this context we consider $\mathcal{M} = \{I, A\}$ and $\mathcal{M}_{(J)} = \{I, D, A, \min\{A^{2^t}, 1\},\ t = 2, \ldots\}$. The operators $\min\{A^{2^t}, 1\}$ allow the model to distinguish regular graphs that order 2 G-invariant networks cannot distinguish, such as the Circular Skip Link graphs.

- **Linear Programming (LP)**. This is not a GNN but the natural linear programming relaxation for graph isomorphism. Namely given a pair graphs with adjacency matrix $A, B \in \{0,1\}^{n\times n}$

$$LP(A, B) = \min \|PA - BP\|_1 \text{ subject to } P\mathbb{1}_n = \mathbb{1}_n,\ P^\intercal \mathbb{1}_n = \mathbb{1}_n,\ P \geq 0.$$

  The natural sigma algebra to consider here is $\sigma(\cup_{A\in\mathcal{X}^{n\times n}}\{LP(A, \cdot)\})$. Two graphs are said to be fractionally isomorphic is $LP(A, B) = 0$ (i.e. the LP cannot distinguish them). [23] showed that two graphs are fractionally isomorphic if and only if they cannot be distinguished by 1-WL.

- **Semidefinite Programming (SDP)**. The semidefinite programming relaxation of quadratic assignment from [32] is based on the following observation: $\|PA - BP\|_F^2 = \|PA\|_F^2 + \|BP\|_F^2 - 2\,\mathrm{trace}(PAP^\top B^\top)$ and $\mathrm{trace}(\mathrm{vec}(P)\,\mathrm{vec}(P)^\top A \otimes B^\top)$ where $\otimes$ is the Kronecker product operator and vec takes an $n \times n$ matrix and flattens it into an $n^2 \times 1$ vector. The resulting semidefinite relaxation considers the vector $x^\top := [1, \mathrm{vec}(P)^\top]$ and relaxes the rank 1 matrix $xx^\top$ into a positive semidefinite matrix. By including the constraints corresponding to the LP in $xx^\top$ one makes sure that solution of the SDP is always in the feasible set of the LP, therefore the LP is less expressive than the SDP.
- **Sum-of-Squares (SoS) hierarchy**. One can consider the hierarchy of relaxations coming from sum-of-squares (SoS). In the context of graph isomorphism, it is known that graph isomorphism is a hard problem for this hierarchy [22]. In particular the Lasserre/SoS hierarchy requires $2^{\Omega}(n)$ to solve graph isomorphism (in the same sense that $o(n)$-WL fails to solve graph isomorphism [4]).
- **Spectral methods**. If we consider the function that takes a graph and outputs the set of eigenvalues of its adjacency matrix, such function is permutation invariant. A priori one may think that such function, being highly non-linear, is more expressive than any form message passing GNN. In fact, regular graphs are not distinguished by 1-WL or order 2 G-invariant networks and may be distinguished by their eigenvalues (like the Circular Skip Link graphs). However, 1-WL and this particular spectral method are not comparable (a simple example is provided in Figure 2 of [23]).

D Graph G-invariant Networks with maximum tensor order 2

In this section we prove Theorem 7 that says that graph G-invariant Networks with tensor order 2 cannot distinguish between non-isomorphic regular graphs with the same degree.

First, we need to state our definition of the order-2 Graph G-invariant Networks. In general, given $G \in \mathbb{R}^{n\times n}$, we let $A^{(0)} = G$, $d^{(0)} = 1$, and

$$A^{(t+1)} = \sigma(L^{(t)}(A^{(t)}))$$

and outputs $m \circ h \circ A^{(L)}$, where each $L^{(t)}$ is an equivariant linear layer from $\mathbb{R}^{n\times n\times d^{(t)}}$ to $\mathbb{R}^{n\times n\times d^{(t+1)}}$, σ is a point-wise activation function, h is an invariant linear layer from $\mathbb{R}^{n\times n}$ to $\mathbb{R}$, and m is an MLP.

$d^{(t)}$ is the feature dimension in layer t, interpreted as the dimension of the hidden state attached to each pair of nodes. For simplicity of notations, in the following proof we assume that $d^{(t)} = 1, \forall t = 1, ..., L$, and thus each $A^{(t)}$ is essentially a matrix. The following results can be extended to the cases where $d^{(t)} > 1$, by adding more subscripts in the proof.

Given an unweighted graph G, let $E \subseteq [n]^2$ be the edge set of G, i.e., $(u, v) \in E$ if $u \neq v$ and $G_{uv} = 1$; set $S \subseteq [n]^2$ to be $\{(u, u)\}_{u\in[n]^2}$; and let $N = [n]^2 \setminus (E \cup S)$. Thus, $E \cup N \cup S = [n]^2$.

Lemma 5. *Let G, G' be the adjacency matrices of two unweighted regular graphs with the same degree d, and let $A^{(t)}, E, N, S$ and $A'^{(t)}, E', N', S'$ be defined as above for G and G', respectively. Then $\forall n \leq L, \exists \xi_1^{(t)}, \xi_2^{(t)}, \xi_3^{(t)} \in \mathbb{R}$ such that $A_{uv}^{(t)} = \xi_1^{(t)} \mathbb{1}_{(u,v)\in E} + \xi_2^{(t)} \mathbb{1}_{(u,v)\in N} + \xi_3^{(t)} \mathbb{1}_{(u,v)\in S}$, and $A'^{(t)}_{uv} = \xi_1^{(t)} \mathbb{1}_{(u,v)\in E'} + \xi_2^{(t)} \mathbb{1}_{(u,v)\in N'} + \xi_3^{(t)} \mathbb{1}_{(u,v)\in S'}$*

Proof. We prove this lemma by induction. For $t = 0$, $A^{(0)} = G$ and $A'^{(0)} = G'$. Since the graph is unweighted, $G_{uv} = 1$ if $u \neq v$ and $(u, v) \in E$, and 0 otherwise. Similar is true for G'. Therefore, we can set $\xi_1^{(0)} = 1$ and $\xi_2^{(0)} = \xi_3^{(0)} = 0$.

Next, we consider the inductive steps. Assume that the conditions in the lemma are satisfied for layer $t - 1$. To simplify the notation, we use A, A' to stand for $A^{(t-1)}, A'^{(t-1)}$, and we assume to satisfy the inductive hypothesis with ξ_1, ξ_2 and ξ_3. We thus want to show that if L is any equivariant linear, then $\sigma(L(A)), \sigma(L(A'))$ also satisfies the inductive hypothesis. Also, in the following, we use p_1, p_2, q_1, q_2 to refer to nodes, a, b to refer to pairs of nodes, λ to refer to any equivalence class of 2-tuples (i.e. pairs) of nodes, and μ to refer to any equivalence class of 4-tuples of nodes.

$\forall a = (p_1, p_2), b = (q_1, q_2) \in [n]^2$, let $\mathcal{E}(a, b)$ denote the equivalence class of 4-tuples containing (p_1, p_2, q_1, q_2), and let $\mathcal{E}(b)$ represent the equivalence class of 2-tuples containing (q_1, q_2). Two 4-tuples $(u, v, w, x), (u', v', w', x')$ are considered equivalent if $\exists \pi \in S_n$ such that $\pi(u) = u', \pi(v) = v', \pi(w) = w', \pi(x) = x'$. Similarly is equivalence between 2-tuples defined. By equation 9(b) in [17], using the notations of T, B, C, w, β defined there, L is described by, given A as an input as b as the subscript index on the output,

$$\begin{aligned} L(A)_b &= \sum_{a=(p_1,p_2)=(1,1)}^{(n,n)} T_{a,b} A_a + Y_b \\ &= \sum_{a,\mu} w_\mu B^\mu_{a,b} A_a + \sum_\lambda \beta_\lambda C^\lambda_b \\ &= \sum_\mu \Big(\sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu}} A_a \Big) w_\mu + \beta_{\mathcal{E}(b)} \end{aligned} \tag{3}$$

First, let

$$S^b_\mu = \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu}} A_a$$

By the inductive hypothesis,

$$\begin{aligned} S^b_\mu &= \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in E}} A_a + \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in N}} A_a + \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in S}} A_a \\ &= \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in E}} \xi_1 + \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in N}} \xi_2 + \sum_{\substack{a \in [n]^2 \\ (a,b) \in \mu \\ a \in S}} \xi_3 \\ &= m_E(b, \mu)\xi_1 + m_N(b, \mu)\xi_2 + m_S(b, \mu)\xi_3 \end{aligned} \tag{4}$$

where $m_E(b, \mu)$ is defined as the total number of distinct $a \in [n]^2$ that satisfies $(a, b) \in \mu$ and $a \in E$, and similarly for $m_N(b, \mu)$ and $m_S(b, \mu)$. Formally, for example, $m_E(b, \mu) = card\{a \in [n]^2 : (a, b) \in \mu, a \in E\}$.

Since $E \cup N \cup S = [n]^2$, b belongs to one of E, N and S. Thus, let $\tau(b) = E$ if $b \in E$, $\tau(b) = N$ if $b \in N$ and $\tau(b) = S$ if $b \in S$. It turns out that if A is the adjacency matrix of a undirected regular graph with degree d, then $m_E(b, \mu), m_N(b, \mu), m_S(b, \mu)$ can be instead written (with an abuse of notation) as $m_E(\tau(b), \mu), m_N(\tau(b), \mu), m_S(\tau(b), \mu)$, meaning that for a fixed μ, the values of m_E, m_N and m_S only depend on which of the three sets (E, N or S) b is in, and changing b to a different member in the set $\tau(b)$ won't change the three numbers. In fact, for each $\tau(b)$ and μ, the three numbers can be computed as functions of n and d using simple combinatorics, and their values are seen in the three tables 2, 3 and 4. An illustration of these numbers is given in Figure D.

Therefore, we have $L(A)_b = \sum_\mu w_\mu(m_E(\tau(b), \mu) + m_N(\tau(b), \mu) + m_S(\tau(b), \mu)) + \beta_{\mathcal{E}(b)}$. Moreover, notice that $\tau(b)$ determines $\mathcal{E}(b)$: if $\tau(b) = E$ or N, then $\mathcal{E}(b) = \mathcal{E}(1, 2)$; if $\tau(b) = S$, then $\mathcal{E}(b) = \mathcal{E}(1, 1)$. Hence, we can write $\beta_{\tau(b)}$ instead of $\beta_{\mathcal{E}(b)}$ without loss of generality. Then in particular, this means that $L(A)_b = L(A)_{b'}$ if $\tau(b) = \tau(b')$. Therefore, $L(A)_b = \overline{\xi}_1 \mathbb{1}_{b \in E} + \overline{\xi}_2 \mathbb{1}_{b \in N} + \overline{\xi}_3 \mathbb{1}_{b \in S}$, where $\overline{\xi}_1 = \sum_\mu w_\mu(m_E(E, \mu) + m_N(E, \mu) + m_S(E, \mu)) + \beta_E$, $\overline{\xi}_2 = \sum_\mu w_\mu(m_E(N, \mu) + m_N(N, \mu) + m_S(N, \mu)) + \beta_N$, and $\overline{\xi}_3 = \sum_\mu w_\mu(m_E(S, \mu) + m_N(S, \mu) + m_S(S, \mu)) + \beta_S$.

Similarly, $L(A')_b = \overline{\xi'}_1 \mathbb{1}_{b \in E'} + \overline{\xi'}_2 \mathbb{1}_{b \in N'} + \overline{\xi'}_3 \mathbb{1}_{b \in S'}$. But importantly, $\forall$ equivalence class of 4-tuples, μ, and $\forall \lambda_1, \lambda_2 \in \{E, N, S\}, m_{\lambda_1}(\lambda_2, \mu) = m'_{\lambda_1}(\lambda_2, \mu)$, as both of them can be obtained from the same entry of the same table. Therefore, $\overline{\xi}_1 = \overline{\xi'}_1, \overline{\xi}_2 = \overline{\xi'}_2, \overline{\xi}_3 = \overline{\xi'}_3$.

Finally, let $\xi^*_1 = \sigma(\overline{\xi}_1), \xi^*_2 = \sigma(\overline{\xi}_2)$, and $\xi^*_3 = \sigma(\overline{\xi}_3)$. Then, there is $\sigma(L(A))_b = \xi^*_1 \mathbb{1}_{b \in E} + \xi^*_2 \mathbb{1}_{b \in N} + \xi^*_3 \mathbb{1}_{b \in S}$, and $\sigma(L(A'))_b = \xi^*_1 \mathbb{1}_{b \in E'} + \xi^*_2 \mathbb{1}_{b \in N'} + \xi^*_3 \mathbb{1}_{b \in S'}$, as desired. □

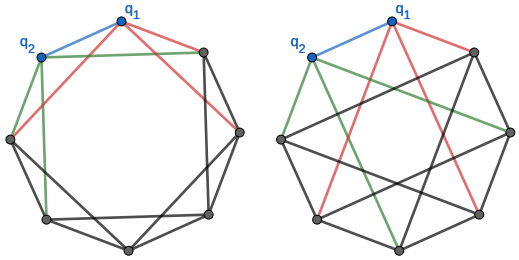

Figure 4: $m_E(E, \mathcal{E}(1,2,3,4))$, $m_E(E, \mathcal{E}(1,2,3,2))$, $m_E(E, \mathcal{E}(1,2,3,1))$, $m_E(E, \mathcal{E}(1,2,2,3))$ and $m_E(E, \mathcal{E}(1,2,1,3))$ of $G_{8,2}$ and $G_{8,3}$. In either graph, twice the total number of black edges equal $m_E(E, \mathcal{E}(1,2,3,4)) = 18$ (it is twice because each undirected edge corrspond to two pairs (p_1, p_2) and (p_2, p_1), which combined with (q_1, q_2) both belongs to $\mathcal{E}(1,2,3,4)$); the total number of of red edges, 3, equals both $m_E(E, \mathcal{E}(1,2,2,3))$ and $m_E(E, \mathcal{E}(1,2,1,3))$; the total number of green edges, also 3, equals both $m_E(E, \mathcal{E}(1,2,3,2))$, $m_E(E, \mathcal{E}(1,2,3,1))$.

μ	$m_E(E,\mu)$	$m_E(N,\mu)$	$m_E(S,\mu)$
(1, 2, 3, 4)	$(n-4)d+2$	$(n-4)d$	0
(1, 1, 2, 3)	0	0	0
(1, 2, 2, 3)	$d-1$	d	0
(1, 2, 1, 3)	$d-1$	d	0
(1, 2, 3, 2)	$d-1$	d	0
(1, 2, 3, 1)	$d-1$	d	0
(1, 1, 1, 2)	0	0	0
(1, 1, 2, 1)	0	0	0
(1, 2, 1, 2)	1	0	0
(1, 2, 2, 1)	1	0	0
(1, 2, 3, 3)	0	0	$(n-2)d$
(1, 1, 2, 2)	0	0	0
(1, 2, 2, 2)	0	0	d
(1, 2, 1, 1)	0	0	d
(1, 1, 1, 1)	0	0	0
Total	nd	nd	nd

Table 2: m_E

Since h is an invariant function, h acting on $A^{(L)}$ essentially computes the sum of all the diagonal terms (i.e., for $b \in S$) and the sum of all the off-diagonal terms (i.e., for $b \in E \cup N$) of $A^{(L)}$ separately and then adds the two sums with two weights. If G, G' are regular graphs with the same degree, then $|E| = |E'|, |S| = |S'|$ and $|N| = |N'|$. Therefore, by the lemma, there is $h(A^{(L)}) = h(A'^{(L)})$, and as a consequence $m(h(A^{(L)})) = m(h(A'^{(L)}))$.

μ	$m_N(E,\mu)$	$m_N(N,\mu)$	$m_N(S,\mu)$
(1, 2, 3, 4)	$(n-4)(n-d-1)$	$(n-4)(n-d-1)+2$	0
(1, 1, 2, 3)	0	0	0
(1, 2, 2, 3)	$n-d-1$	$n-d-2$	0
(1, 2, 1, 3)	$n-d-1$	$n-d-2$	0
(1, 2, 3, 2)	$n-d-1$	$n-d-2$	0
(1, 2, 3, 1)	$n-d-1$	$n-d-2$	0
(1, 1, 1, 2)	0	0	0
(1, 1, 2, 1)	0	0	0
(1, 2, 1, 2)	0	1	0
(1, 2, 2, 1)	0	1	0
(1, 2, 3, 3)	0	0	$(n-2)(n-d-1)$
(1, 1, 2, 2)	0	0	0
(1, 2, 2, 2)	0	0	$n-d-1$
(1, 2, 1, 1)	0	0	$n-d-1$
(1, 1, 1, 1)	0	0	0
Total	$n(n-d-1)$	$n(n-d-1)$	$n(n-d-1)$

Table 3: m_N

μ	$m_S(E,\mu)$	$m_S(N,\mu)$	$m_S(S,\mu)$
(1, 2, 3, 4)	0	0	0
(1, 1, 2, 3)	$n-2$	$n-2$	0
(1, 2, 2, 3)	0	0	0
(1, 2, 1, 3)	0	0	0
(1, 2, 3, 2)	0	0	0
(1, 2, 3, 1)	0	0	0
(1, 1, 1, 2)	1	1	0
(1, 1, 2, 1)	1	1	0
(1, 2, 1, 2)	0	0	0
(1, 2, 2, 1)	0	0	0
(1, 2, 3, 3)	0	0	0
(1, 1, 2, 2)	0	0	$n-1$
(1, 2, 2, 2)	0	0	0
(1, 2, 1, 1)	0	0	0
(1, 1, 1, 1)	0	0	1
Total	n	n	n

Table 4: m_S

E Specific GNN Architectures

In section 6, we show experiments on synthetic and real datasets with several related architectures. Here are some explanations for them.

- **sGNN-i**: sGNNs with operators from family $\{I, D, \min(A^{2^0}, 1), \ldots, \min(A^{2^{i-1}}, 1)\}, i \in \{1, 2, 5\}$. In our experiments, the $sGNN$ models have 5 layers and hidden layer dimension (i.e. d^k) 64. They are trained using the Adam optimizer with learning rate 0.01.
- **LGNN**: Line Graph Neural Networks proposed by [5]. In our experiments, the $sGNN$ models have 5 layers and hidden layer dimension (i.e. d^k) 64. They are trained using the Adam optimizer with learning rate 0.01.
- **GIN**: Graph Isomorphism Network by [30]. We took their performance results on the IMDB datasets reported in [30], and their performance results on the Circular Skip Link graphs experiments reported in [20] .
- **RP-GIN**: Graph Isomorphism Network combined with Relational pooling by [20]. We took the reported results reported in [20] for the Circular Skip Link graphs experiment.
- **Order-2 Graph G-invariant Networks**: G-invariant networks based on [17] and [18], as implemented in https://github.com/Haggaim/InvariantGraphNetworks.

- **Ring-GNN**: As defined in the main text. The architecture (number of hidden layers, feature dimensions) is taken to be the same os the Order-2 Graph G-invariant Networks. For the experiments on the IMDB datasets, each $k_1^{(t)}$ is initialized independently under $\mathcal{N}(0, 1)$, and each $k_2^{(t)}$ is initialized independently under $\mathcal{N}(0, 0.01)$. They are trained using the Adam optimizer with learning rate 0.00001. The initialization of $k_2^{(t)}$ and the learning rate were manually tuned, following the heuristic that Ring-GNN reduces to Order-2 Graph G-invariant Networks when $k_2^{(t)} = 0$, and that since Ring-GNN added more operators, a smaller learning rate is likely more appropriate.
- **Ring-GNN-SVD**: Compared with above Ring-GNN model, a Singular Value Decomposition layer is added between Ring layers and fully-connected layers. SVD layer takes as input batch-size$\times$channels matrices and as output batch-size$\times$channels$\times$5 top eigenvalues. Considering computation complexity and condition numbers, this model has only two Ring layers and careful initialization. For IMDB datasets, Ring layers have numbers of channels in $\{16, 32\}$ and the model is trained using Adam optimizer with learning rate of 0.001 for 350 epochs. For CSL dataset, Ring layers have numbers of channels in $\{4, 8\}$ and the model is trained using Adam optimizer with learning rate of 0.001 for 1000 epochs. In both cases, each $k_1^{(t)}$ is initializated independently under $\mathcal{N}(0, 0.5)$ and each $k_2^{(t)}$ is initializated independently under $\mathcal{N}(0, 0.005)$. It is also noted, since often easily dropping into ill condition when using back propagation of SVD, we clip gradient values when training. Moreover, from prospective of computation resources, Nvidia V100 and P40 are much more numerically robust than 1080Ti and CPU in this task.

For the experiments with Circular Skip Links graphs, each model is trained and evaluated using 5-fold cross-validation. For Ring-GNN, in particular, we performed training + cross-validation 20 times with different random seeds.