ON GRAPH NEURAL NETWORKS VERSUS GRAPH-AUGMENTED MLPS

Lei Chen*, **Zhengdao Chen***
Courant Institute of Mathematical Sciences
New York University, New York, NY
{lc3909, zc1216}@nyu.edu

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

ABSTRACT

From the perspectives of expressive power and learning, this work compares multi-layer Graph Neural Networks (GNNs) with a simplified alternative that we call Graph-Augmented Multi-Layer Perceptrons (GA-MLPs), which first augments node features with certain multi-hop operators on the graph and then applies learnable node-wise functions. From the perspective of graph isomorphism testing, we show both theoretically and numerically that GA-MLPs with suitable operators can distinguish almost all non-isomorphic graphs, just like the Weisfeiler-Lehman (WL) test and GNNs. However, by viewing them as node-level functions and examining the equivalence classes they induce on rooted graphs, we prove a separation in expressive power between GA-MLPs and GNNs that grows exponentially in depth. In particular, unlike GNNs, GA-MLPs are unable to count the number of attributed walks. We also demonstrate via community detection experiments that GA-MLPs can be limited by their choice of operator family, whereas GNNs have higher flexibility in learning.

1 INTRODUCTION

While multi-layer Graph Neural Networks (GNNs) have gained popularity for their applications in various fields, recently authors have started to investigate what their true advantages over baselines are, and whether they can be simplified. On one hand, GNNs based on neighborhood-aggregation allows the combination of information present at different nodes, and by increasing the depth of such GNNs, we increase the size of the receptive field. On the other hand, it has been pointed out that deep GNNs can suffer from issues including over-smoothing, exploding or vanishing gradients in training as well as bottleneck effects (Kipf & Welling, 2016; Li et al., 2018; Luan et al., 2019; Oono & Suzuki, 2020; Rossi et al., 2020; Alon & Yahav, 2020).

Recently, a series of models have attempted at relieving these issues of deep GNNs while retaining their benefit of combining information across nodes, using the approach of firstly augmenting the node features by propagating the original node features through powers of graph operators such as the (normalized) adjacency matrix, and secondly applying a node-wise function to the augmented node features, usually realized by a Multi-Layer Perceptron (MLP) (Wu et al., 2019; NT & Maehara, 2019; Chen et al., 2019a; Rossi et al., 2020). Because of the usage of graph operators for augmenting the node features, we will refer to such models as *Graph-Augmented MLPs (GA-MLPs)*. These models have achieved competitive performances on various tasks, and moreover enjoy better scalability since the augmented node features can be computed during preprocessing (Rossi et al., 2020). Thus, it becomes natural to ask what advantages GNNs have over GA-MLPs.

In this work, we ask whether GA-MLPs sacrifice expressive power compared to GNNs while gaining these advantages. A popular measure of the expressive power of GNNs is their ability to distinguish non-isomorphic graphs (Hamilton et al., 2017; Xu et al., 2019; Morris et al., 2019). In our work, besides studying the expressive power of GA-MLPs from the viewpoint of graph isomorphism tests, we propose a new perspective that better suits the setting of node-prediction tasks: we analyze the

*Equal contributions. Code available at https://github.com/leichen2018/GNN_vs_GAMLP.

expressive power of models including GNNs and GA-MLPs as node-level functions, or equivalently, as functions on rooted graphs. Under this perspective, we prove an exponential-in-depth gap between the expressive powers of GNNs and GA-MLPs. We illustrate this gap by finding a broad family of user-friendly functions that can be provably approximated by GNNs but not GA-MLPs, based on counting attributed walks on the graph. Moreover, via the task of community detection, we show a lack of flexibility of GA-MLPs, compared to GNNs, to learn the best operators to use.

In summary, our main contributions are:

- Finding graph pairs that several GA-MLPs cannot distinguish while GNNs can, but also proving there exist simple GA-MLPs that distinguish almost all non-isomorphic graphs.
- From the perspective of approximating node-level functions, proving an exponential gap between the expressive power of GNNs and GA-MLPs in terms of the equivalence classes on rooted graphs that they induce.
- Showing that the functions that count a particular type of attributed walk among nodes can be approximated by GNNs but not GA-MLPs both in theory and numerically.
- Through community detection tasks, demonstrating that GNNs have higher flexibility in learning than GA-MLPs due to the fixed choice of the operator family in the latter.

2 Related Works

Depth in GNNs Kipf & Welling (2016) observe that the performance of Graph Convolutional Networks (GCNs) degrade as the depth grows too large, and the best performance is achieved with 2 or 3 layers. Along the spectral perspective on GNNs (Bruna et al., 2013; Defferrard et al., 2016; Bronstein et al., 2017; NT & Maehara, 2019), Li et al. (2018) and Wu et al. (2019) explain the failure of deep GCNs by the over-smoothing of the node features. Oono & Suzuki (2020) show an exponential loss of expressive power as the depth in GCNs increases in the sense that the hidden node states tend to converge to Laplacian sub-eigenspaces as the depth increases to infinity. Alon & Yahav (2020) show an over-squashing effect of deep GNNs, in the sense that the width of the hidden states needs to grow exponentially in the depth in order to retain all information about long-range interactions. In comparison, our work focuses on more general GNNs based on neighborhood-aggregation that are not limited in the hidden state widths, and demonstrates the their *advantage* in expressive power compared to GA-MLP models at finite depth, in terms of distinguishing rooted graphs for node-prediction tasks. On the other hand, there exist examples of useful deep GNNs. Chen et al. (2019b) apply 30-layer GNNs for community detection problems, which uses a family of multi-scale operators as well as normalization steps (Ioffe & Szegedy, 2015; Ulyanov et al., 2016). Recently, Li et al. (2019; 2020a) and Chen et al. (2020a) build deeper GCN architectures with the help of various residual connections (He et al., 2016) and normalization steps to achieve impressive results in standard datasets, which further highlights the need to study the role of depth in GNNs. Gong et al. (2020) propose geometrically principled connections, which improve upon vanilla residual connections on graph- and mesh-based tasks.

Existing GA-MLP-type models Motivated by better understanding GNNs as well as enhancing computational efficiency, several models of the GA-MLP type have been proposed and they achieve competitive performances on various datasets. Wu et al. (2019) propose the Simple Graph Convolution (SGC), which removes the intermediary weights and nonlinearities in GCNs. Chen et al. (2019a) propose the Graph Feature Network (GFN), which further adds intermediary powers of the normalized adjacency matrix to the operator family and is applied to graph-prediction tasks. NT & Maehara (2019) propose the Graph Filter Neural Networks (gfNN), which enhances the SGC in the final MLP step. Rossi et al. (2020) propose Scalable Inception Graph Neural Networks (SIGNs), which augments the operator family with Personalized-PageRank-based (Klicpera et al., 2018; 2019) and triangle-based (Monti et al., 2018; Chen et al., 2019b) adjacency matrices.

Expressive Power of GNNs Xu et al. (2019) and Morris et al. (2019) show that GNNs based on neighborhood-aggregation are no more powerful than the Weisfeiler-Lehman (WL) test for graph isomorphism (Weisfeiler & Leman, 1968), in the sense that these GNNs cannot distinguish between any pair of non-isomorphic graphs that the WL test cannot distinguish. They also propose models that match the expressive power of the WL test. Since then, many attempts have been made to build

GNN models whose expressive power are not limited by WL (Morris et al., 2019; Maron et al., 2019a; Chen et al., 2019c; Morris & Mutzel, 2019; You et al., 2019; Bouritsas et al., 2020; Li et al., 2020b; Flam-Shepherd et al., 2020; Sato et al., 2019; 2020). Other perspectives for understanding the expressive power of GNNs include function approximation (Maron et al., 2019b; Chen et al., 2019c; Keriven & Peyré, 2019), substructure counting (Chen et al., 2020b), Turing universality (Loukas, 2020) and the determination of graph properties (Garg et al., 2020). Sato (2020) provides a survey on these topics. In this paper, besides studying the expressive power of GA-MLPs along the line of graph isomorphism tests, we propose a new perspective of approximating functions on rooted graphs, which is motivated by node-prediction tasks, and show a gap between GA-MLPs and GNNs that grows exponentially in the size of the receptive field in terms of the equivalence classes that they induce on rooted graphs.

3 BACKGROUND

3.1 NOTATIONS

Let $G = (V, E)$ denote a graph, with V being the vertex set and E being the edge set. Let n denote the number of nodes in G, $A \in \mathbb{R}^{n\times n}$ denote the *adjacency matrix*, $D \in \mathbb{R}^{n\times n}$ denote the diagonal *degree matrix* with $D_{ii} = d_i$ being the degree of node i. We call $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ the *(symmetrically) normalized adjacency matrix*, and $D^{-\alpha}AD^{-\beta}$ a *generalized normalized adjacency matrix* for any $\alpha, \beta \in \mathbb{R}$. Let $X \in \mathbb{R}^{n\times d}$ denote the matrix of node features, where X_i denotes the d-dimensional feature that node i possesses. For a node $i \in V$, let $\mathcal{N}(i)$ denote the set of neighbors of i. We assume that the edges do not possess features. In a node prediction task, the labels are given by $Y \in \mathbb{R}^n$.

For a positive integer K, we let $[K] = \{1, ..., K\}$. We use $\{...\}_m$ to denote a multiset, which allows repeated elements. We say a function $f(K)$ is *doubly-exponential* in K if $\log\log f(K)$ is polynomial in K, and *poly-exponential* in K if $\log f(K)$ is polynomial in K, as K tends to infinity.

3.2 GRAPH NEURAL NETWORKS (GNNS)

Following the notations in Xu et al. (2019), we consider K-layer GNNs defined generically as follows. For $k \in [K]$, we compute the hidden node states $H \in \mathbb{R}^{n\times d^{(k)}}$ iteratively as

$$M_i^{(k)} = \text{AGGREGATE}^{(k)}(\{H_j^{(k-1)} : j \in \mathcal{N}(i)\}) \,, \quad H_i^{(k)} = \text{COMBINE}^{(k)}(H_i^{(k-1)}, M_i^{(k)}) \,, \quad (1)$$

where we set $H^{(0)} = X$ to be the node features. If a graph-level output is desired, we finally let

$$Z_G = \text{READOUT}(\{H_i^{(K)} : i \in V\}) \,, \quad (2)$$

Different choices of the trainable COMBINE, AGGREGATE and READOUT functions result in different GNN models, though usually AGGREGATE and READOUT are chosen to be permutation-invariant. As graph-level functions, it is shown in Xu et al. (2019) and Morris et al. (2019) that the maximal expressive power of models of this type coincides with running K iterations of the WL test for graph isomorphism, in the sense that any two non-isomorphic graphs that cannot be distinguished by the latter cannot be distinguished by the K-layer GNNs, either. For this reason, we will not distinguish between GNN and WL in discussions on expressive powers.

3.3 GRAPH-AUGMENTED MULTI-LAYER PECEPTRONS (GA-MLPS)

GA-MLPs are models that consist of two steps - first augmenting the node features with some operators based on the graph topology, and then applying a node-wise learnable function. Below we focus on using linear graph operators to augment the node features, while an extension of the definition as well as some of the theoretical results in Section 5 to GA-MLPs using general graph operators is given in Appendix A. Let $\Omega = \{\omega_1(A), ..., \omega_K(A)\} \subseteq \mathbb{R}^{n\times n}$ be a set of (usually multi-hop) linear operators that are functions of the adjacency matrix, A. Common choices of the operators are powers of the (normalized) adjacency matrix, and several particular choices of Ω that give rise to existing GA-MLP models are listed in Appendix B. In its general form, a GA-MLP first computes a series of augmented features via

$$\tilde{X}_k = \omega_k(A) \cdot \varphi(X) \in \mathbb{R}^{n\times \tilde{d}} \,, \quad (3)$$

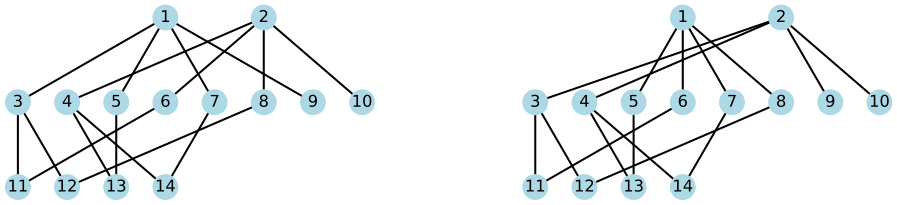

Figure 1: A pair of graphs that can be distinguished by 2 iterations of the WL test but not by GA-MLPs with $\Omega \subseteq \{A^k : k \in \mathbb{N}\}$, as proved in Appendix K.

with $\varphi : \mathbb{R}^d \to \mathbb{R}^{\tilde{d}}$ being a learnable function acting as a feature transformation applied to each node separately. It can be realized by an MLP, e.g. $\varphi(X) = \sigma(XW_1)W_2$, where σ is a nonlinear activation function and W_1, W_2 are trainable weight matrices of suitable dimensions. Next, the model concatenates $\tilde{X}_1, ..., \tilde{X}_K$ into $\tilde{X} = [\tilde{X}_1, ..., \tilde{X}_K] \in \mathbb{R}^{n \times (K\tilde{d})}$, and computes

$$Z = \rho(\tilde{X}) \in \mathbb{R}^{n \times d'} , \tag{4}$$

where $\rho : \mathbb{R}^{K\tilde{d}} \to \mathbb{R}^{d'}$ is also a learnable node-wise function, again usually realized by an MLP. If a graph-level output is desired, we can also add a READOUT function as in (2).

A simplified version of the model sets φ to be the identity function, in which case (3) and (4) can be written together as

$$Z = \rho([\omega_1(A) \cdot X, ..., \omega_K(A) \cdot X]) \tag{5}$$

Such a simplification improves computational efficiency since the matrix products $\omega_k(A) \cdot X$ can be pre-computed before training (Rossi et al., 2020). Since we are mostly interested in an upper bounds on the expressive power of GA-MLPs, we will work with the more general update rule (3) in this paper, but the lower-bound result in Proposition 2 remains valid even when we restrict to the subset of models where φ is taken to be the identity function.

4 Expressive Power as Graph Isomorphism Tests

We first study the expressive power of GA-MLPs via their ability to distinguish non-isomorphic graphs. It is not hard to see that when $\Omega = \{I, \tilde{A}, ..., \tilde{A}^K\}$, where $\tilde{A} = D^{-\alpha}AD^{-\beta}$ for any $\alpha, \beta \in \mathbb{R}$ generalizes the normalized adjacency matrix, this is upper-bounded by the power of $K+1$ iterations of WL. We next ask whether it can fall strictly below. Indeed, for two common choices of Ω, we can find concrete examples: 1) If Ω consists of integer powers of any normalized adjacency matrix of the form $D^{-\alpha}AD^{-(1-\alpha)}$ for some $\alpha \in [0, 1]$, then it is apparent that the GA-MLP cannot distinguish any pair of *regular graphs* with the same size but different node degrees; 2) If Ω consists of integer powers of the adjacency matrix, A, then the model cannot distinguish between the pair of graphs shown in Figure 1, which can be distinguished by 2 iterations of the WL test. The proof of the latter result is given in Appendix K. Together, we summarize the results as:

Proposition 1. *If $\Omega \subseteq \{\tilde{A}^k : k \in \mathbb{N}\}$, with either $\tilde{A} = A$ or $\tilde{A} = D^{-\alpha}AD^{-(1-\alpha)}$ for some $\alpha \in [0, 1]$, there exists a pair of graphs which can be distinguished by GNNs but not this GA-MLP.*

Nonetheless, if we focus on not particular counterexamples but rather the average performance in distinguishing random graphs, it is not hard for GA-MLPs to reach the same level as WL, which is known to distinguish almost all pairs of random graphs under a uniform distribution (Babai et al., 1980). Specifically, building on the results in Babai et al. (1980), we prove in Appendix F that:

Proposition 2. *For all $n \in \mathbb{N}_+$, $\exists \alpha_n > 0$ such that any GA-MLP that has $\{D, AD^{-\alpha_n}\} \subseteq \Omega$ can distinguish almost all pairs of non-isomorphic graphs of at most n nodes, in the sense that the fraction of graphs on which such a GA-MLP fails to test isomorphism is $1 - o(1)$ as $n \to \infty$.*

The hypothesis that distinguishing non-isomorphic graphs is not difficult on average for either GNNs or GA-MLPs is further supported by the numerical results provided in Appendix C, in which we count the number of equivalence classes that either of them induce on graphs that occur in real-world datasets. This further raises the question of whether graph isomorphism tests along suffice as a

criterion for comparing the expressive power of models on graphs, which leads us to the explorations in the next section.

Lastly, we remark that with suitable choices of operators in Ω, it is possible for GA-MLPs to go beyond the power of WL. For example, if Ω contains the *power graph adjacency matrix* introduced in Chen et al. (2019b), $\min(A^2, 1)$, then the GA-MLP can distinguish between a hexagon and a pair of triangles, which WL cannot distinguish.

5 Expressive Power as Functions on Rooted Graphs

To study the expressive power beyond graph isomorphism tests, we consider the setting of node-wise prediction tasks, for which the final readout step (2) is dropped in both GNNs and GA-MLPs. Whether the learning setup is transductive or inductive, we can consider the models as functions on *rooted graphs*, or *egonets* (Preciado & Jadbabaie, 2010), which are graphs with one node designated as the root $\{i_1, ..., i_n\}$ is a set of nodes in the graphs $\{G_1, ..., G_n\}$ (not necessarily distinct) and with node-level labels $\{Y_{i_1}, ..., Y_{i_n}\}$ known during training, respectively, then the goal is to fit a function to the input-output pairs $(G_n^{[i_n]}, Y_{i_n})$, where we use $G^{[i]}$ to denote the rooted graph with G being the graph and the node i in G being the root. Thus, we can evaluate the expressive power of GNNs and GA-MLPs by their ability to approximate functions on the space of rooted graphs, which we call $\mathcal{E}$.

To do so, we introduce a notion of induced equivalence relations on $\mathcal{E}$, analogous to the equivalence relations on $\mathcal{G}$ introduced in Appendix C. Given a family of functions $\mathcal{F}$ on $\mathcal{E}$, we can define an equivalence relation $\simeq_{\mathcal{E};\mathcal{F}}$ among all rooted graphs such that $\forall G^{[i]}, G'^{[i']} \in \mathcal{E}$, $G^{[i]} \simeq_{\mathcal{E};\mathcal{F}} G'^{[i']}$ if and only if $\forall f \in \mathcal{F}$, $f(G^{[i]}) = f(G'^{[i']})$. By examining the number and sizes of the induced equivalence classes of rooted graphs, we can evaluate the relative expressive power of different families of functions on $\mathcal{E}$ in a quantitative way.

In the rest of this section, we assume that the node features belong to a finite alphabet $\mathcal{X} \subseteq \mathbb{N}$ and all nodes have degree at most $m \in \mathbb{N}_+$. Firstly, GNNs are known to distinguish neighborhoods up to the *rooted aggregation tree*, which can be obtained by unrolling the neighborhood aggregation steps in the GNNs as well as the WL test (Xu et al., 2019; Morris et al., 2019; Garg et al., 2020). The *depth-K rooted aggregation tree* of a rooted graph $G^{[i]}$ is a depth-K rooted tree with a (possibly many-to-one) mapping from every node in the tree to some node in $G^{[i]}$, where (i) the root of the tree is mapped to node i, and (ii) the children of every node j in the tree are mapped to the neighbors of the node in $G^{[i]}$ to which j is mapped. An illustration of rooted graphs and rooted aggregation trees is given in Figure 4. Hence, each equivalence class in $\mathcal{E}$ induced by the family of all depth-K GNNs consists of all rooted graphs that share the same rooted aggregation tree of depth-K. Thus, to estimate the number of equivalence classes on $\mathcal{E}$ induced by GNNs, we need to estimate the number of possible rooted aggregation trees, which is given by Lemma 3 in Appendix G. Thus, we derive the following lower bound on the number of equivalence classes in $\mathcal{E}$ that depth-K GNNs induce:

Proposition 3. *Assume that $|\mathcal{X}| \geq 2$ and $m \geq 3$. The total number of equivalence classes of rooted graphs induced by GNNs of depth K grows at least doubly-exponentially in K.*

In comparison, we next demonstrate that the equivalence classes induced by GA-MLPs are more coarsened. To see this, let's first consider the example where we take $\Omega = \{I, \tilde{A}, \tilde{A}^2, ..., \tilde{A}^K\}$, in which $\tilde{A} = D^{-\alpha}AD^{-\beta}$ with any $\alpha, \beta \in \mathbb{R}$ is a generalization of the normalized adjacency matrix. From formula (3), by expanding the matrix product, we have

$$(\tilde{A}^k\varphi(X))_i = \sum_{(i_1,...,i_k)\in\mathcal{W}_k(G^{[i]})} d_i^{-\alpha} d_{i_1}^{-(\alpha+\beta)} ... d_{i_{k-1}}^{-(\alpha+\beta)} d_{i_k}^{-\beta} \varphi(X_{i_k}) \,, \tag{6}$$

where we define $\mathcal{W}_k(G^{[i]}) = \{(i_1, ..., i_k) \subseteq V : A_{i,i_1}, A_{i_1,i_2}, ..., A_{i_{k-1},i_k} > 0\}$ to be set of all *walks* of length k in the rooted graph $G^{[i]}$ starting from node i (an illustration is given in Figure 2). Thus, the kth augmented feature of node i, $(\tilde{A}^k\varphi(X))_i$, is completely determined by the number of each "type" of walks in $G^{[i]}$ of length k, where the type of a walk, $(i_1, ..., i_k)$, is determined jointly by the degree multiset, $\{d_{i_1}, ..., d_{i_{k-1}}\}$ as well as the degree and the node feature of the end node, d_{i_k} and X_{i_k}. Hence, to prove an upper bound on the total number of equivalence classes on $\mathcal{E}$ induced by such a GA-MLP, it is sufficient to upper-bound the total number of possibilities of

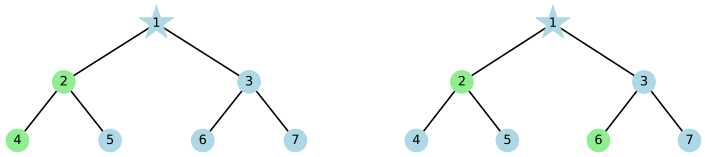

Figure 2: A pair of rooted graphs, $G^{[1]}$ (left) and $G'^{[1]}$ (right), in which blue nodes have node feature 0 and green nodes have node feature 1. They belong to the same equivalence class induced by any GA-MLP with operators that only depend on the graph structure, but different equivalence classes induced by GNNs. In particular, $G^{[1]}$ and $G'^{[1]} \in \mathcal{T}_{2,2,(1,1,3)}$ (defined in Appendix I), and $|\mathcal{W}_2(G^{[1]}; (1,1))| = 1$ whereas $|\mathcal{W}_2(G'^{[1]}; (1,1))| = 0$.

assigning the counts of all types of walks in a rooted graph. This allows us to derive the following result, which we prove in Appendix H:

Proposition 4. *Fix $\Omega = \{I, \tilde{A}, \tilde{A}^2, ..., \tilde{A}^K\}$, where $\tilde{A} = D^{-\alpha}AD^{-\beta}$ for some $\alpha, \beta \in \mathbb{R}$. Then the total number of equivalence classes in $\mathcal{E}$ induced by such GA-MLPs is poly-exponential in K.*

Compared with Proposition 3, this shows that the number of equivalence classes on $\mathcal{E}$ induced by such GA-MLPs is exponentially smaller than that by GNNs. In addition, as the other side of the same coin, these results also indicate the complexity of these hypothesis classes. Building on the results in Chen et al. (2019c; 2020b) on the equivalence between distinguishing non-isomorphic graphs and approximating arbitrary permutation-invariant functions on graphs by neural networks, and by the definition of *VC dimension* (Vapnik & Chervonenkis, 1971; Mohri et al., 2018), we conclude that

Corollary 1. *The VC dimension of all GNNs of K layers as functions on rooted graphs grows at least doubly-exponentially in K; Fixing $\alpha, \beta \in \mathbb{R}$, the VC dimension of all GA-MLPs with $\Omega = \{I, \tilde{A}, \tilde{A}^2, ..., \tilde{A}^K\}$ as functions on rooted graphs is at most poly-exponential in K.*

Meanwhile, for more general operators, we can show that the equivalence classes induced by GA-MLPs are *coarser* than those induced by GNNs at least under some measurements. For instance, the pair of rooted graphs in Figure 2 belong to the same equivalence class induced by any GA-MLP (as we prove in Appendix I) but different equivalence classes induced by GNNs. Rigorously, we characterize such a gap in expressive power by finding certain equivalence classes in $\mathcal{E}$ induced by GA-MLPs that intersect with many equivalence classes induced by GNNs. In particular, we have the following general result, which we prove in Appendix I:

Proposition 5. *If Ω is any family of equivariant linear operators on the graph that only depend on the graph topology of at most K hops, then there exist exponentially-in-K many equivalence classes in $\mathcal{E}$ induced by the GA-MLPs with Ω, each of which intersects with doubly-exponentially-in-K many equivalence classes in $\mathcal{E}$ induced by depth-K GNNs, assuming that $|\mathcal{X}| \geq 2$ and $m \geq 3$. Conversely, in constrast, if $\Omega = \{I, \tilde{A}, \tilde{A}^2, ..., \tilde{A}^K\}$, in which $\tilde{A} = D^{-\alpha}AD^{-\beta}$ with any $\alpha, \beta \in \mathbb{R}$, then each equivalence class in $\mathcal{E}$ induced by depth-$(K+1)$ GNNs is contained in one equivalence class induced by the GA-MLPs with Ω.*

In essence, this result establishes that GA-MLP circuits can express fewer (exponentially fewer) functions than GNNs with equivalent receptive field. Taking a step further, we can find explicit functions on rooted graphs that can be approximated by GNNs but not GA-MLPs. In the framework that we have developed so far, this occurs when the image of each equivalence class in $\mathcal{E}$ induced by GNNs under this function contains a single value, whereas the image of some equivalence class in $\mathcal{E}$ induced by GA-MLPs contains multiple values. Inspired by the proofs of the results above, a natural candidate is the family of functions that count the number of walks of a particular type in the rooted graph. We can establish the following result, which we prove in Appendix J:

Proposition 6. *For any sequence of node features $\{x_k\}_{k \in \mathbb{N}_+} \subseteq \mathcal{X}$, consider the sequence of functions $f_k(G^{[i]}) := |\mathcal{W}_k(G^{[i]}; (x_1, ..., x_k))|$ on $\mathcal{E}$. For all $k \in \mathbb{N}_+$, the image under f_k of every equivalence class in $\mathcal{E}$ induced by depth-k GNNs contains a single value, while for any GA-MLP using equivariant linear operators that only depend on the graph topology, there exist exponentially-in-k many equivalence classes in $\mathcal{E}$ induced by this GA-MLP whose image under f_k contains exponentially-in-k many values.*

In other words, there exist graph instances where the attributed-walk-counting-function f_k takes different values, yet no GA-MLP model can predict them apart – and there are exponentially many

	Cora		Citeseer		Pubmed	
# Nodes	2708		3327		19717	
K	GNN	GA-MLP	GNN	GA-MLP	GNN	GA-MLP
1	37	37	31	31	82	82
2	1589	756	984	506	8059	3762
3	2301	2158	1855	1550	12814	12014
4	2363	2359	2074	2019	12990	12979
5	2365	2365	2122	2115	12998	12998

Table 1: The number of equivalence classes of rooted graphs induced by GNN and GA-MLP on node classification datasets with node features removed.

	Cora		RRG	
Model	Train	Test	Train	Test
GIN	3.98E-6	9.72E-7	3.39E-5	2.61E-4
GA-MLP-A	1.23E-1	1.56E-1	1.75E-2	2.13E-2
GA-MLP-A+	1.87E-2	6.44E-2	1.69E-2	2.13E-2
GA-MLP-$\tilde{A}_{(1)}$	4.22E-1	5.79E-1	1.02E-1	1.58E-1
GA-MLP-$\tilde{A}_{(1)}$+	4.00E-1	5.79E-1	1.12E-1	1.52E-1

Table 2: MSE loss divided by label variance for counting attributed walks on the Cora graph and RRG. The models denoted as "+" contain twice as many powers of the operator.

of these instances as the number of hops increases. This suggests the possibility of lower-bounding the average approximation error for certain functions by GA-MLPs under various random graph families, which we leave for future work.

6 Experiments

The baseline GA-MLP models we consider has operator family $\Omega = \{I, A, ..., A^K\}$ for a certain K, and we call it GA-MLP-A. In Section 6.2 and 6.3, we also consider GA-MLPs with $\Omega = \{I, \tilde{A}_{(1)}, ..., \tilde{A}_{(1)}^K\}$ ($\tilde{A}_{(\epsilon)}$ is defined in Appendix B), denoted as GA-MLP-$\tilde{A}_{(1)}$. For the experiments in Section 6.3, due to the large K as well as the analogy with spectral methods (Chen et al., 2019b), we use instance normalization (Ulyanov et al., 2016). Further details are described in Appendix L.

6.1 Number of equivalence classes of rooted graphs

Motivated by Propositions 3 and 4, we numerically count the number of equivalence classes induced by GNNs and GA-MLPs among the rooted graphs found in actual graphs with node features removed. For depth-K GNNs, we implement a WL-like process with hash functions to map the depth-K egonet associated with each node to a string before comparing across nodes. For GA-MLP-A, we compare the augmented features of each egonet computed via (3). From the results in Table 1, we see that indeed, the number of equivalence classes induced by GA-MLP-A is smaller than that by GNNs, with the highest relative difference occurring at $K = 2$. The contrast is much more visible than their difference in the number of *graph* equivalence classes given in Appendix C.

6.2 Counting attributed walks

Motivated by Proposition 6, we test the ability of GNNs and GA-MLPs to count the number of walks of a particular type in synthetic data. We take graphs from the Cora dataset (with node features removed) as well as generate a random regular graph (RRG) with 1000 nodes and the node degree being 6. We assign node feature *blue* to all nodes with even index and node feature *red* to all nodes with odd index, due to which the node feature is given by 2-dimensional one-hot encoding. On the Cora graph, a node i's label is given by the number of walks of the type *blue*$\to$*blue*$\to$*blue* starting from i. On the RRG, the label is given by the number of walks of the type *blue*$\to$*blue*$\to$*blue*$\to$*blue* starting from i. The number of nodes for training and testing is split as $1000/1708$ for the Cora graph and $300/700$ for the random regular graph. We test four GA-MLP models, two with as many powers of the operator as the walk length and the other two with twice as many operators, and compare their performances against that of the Graph Isomorphism Network (GIN), a GNN model that achieves the expressive power of the WL test (Xu et al., 2019). From Table 2, we see that GIN significantly outperforms GA-MLPs in both training and testing on both graphs, consistent with the theoretical result in Proposition 6 that GNNs can count attributed walks while GA-MLPs cannot. Thus, this points out an intuitive task that lies in the gap of expressive power between GNNs and GA-MLPs.

6.3 Community detection on Stochastic Block Models (SBM)

We use the task of community detection to illustrate another limitation of GA-MLP models: a lack of flexibility to *learn* the family of operators. SBM is a random graph model in which nodes are partitioned into underlying communities and each edge is drawn independently with a probability

that only depends on whether the pair of nodes belong to the same community or not. The task of community detection is then to recover the community assignments from the connectivity pattern. We focus on binary (that is, having two underlying communities) SBM in the sparse regime, where it is known that the hardness of detecting communities is characterized by a signal-to-noise ratio (SNR) that is a function of the in-group and out-group connectivity (Abbe, 2017). We select 5 pairs of in-group and out-group connectivity, resulting in 5 different hardness levels of the task.

Among all different approaches to community detection, spectral methods are particularly worth mentioning here, which usually aim at finding a certain eigenvector of a certain operator that is correlated with the community assignment, such as the second largest eigenvector of the adjacency matrix or the second smallest eigenvector of the Laplacian matrix or the Bethe Hessian matrix (Krzakala et al., 2013). In particular, the Bethe Hessian matrix is known to be asymptotically optimal in the hard regime, provided that a data-dependent parameter is known. Note that spectral methods bear close resemblance to GA-MLPs and GNNs. In particular, Chen et al. (2019b) propose a spectral GNN (sGNN) model for community detection that can be viewed as a learnable generalization of power iterations on a collection of operators. Further details on Bethe Hessian and sGNN are provided in Appendix L.

We first compare two variants of GA-MLP models: GA-MLP-A with $K = 120$, and GA-MLP-H with Ω generated from the Bethe Hessian matrix with the oracle data-dependent parameter also up to $K = 120$. From Figure 3, we see that the latter consistently outperforms the former, indicating the importance of the choice of the operators for GA-MLPs. As reported in Appendix L, replacing A by $\tilde{A}_{(1)}$ yields no improvement in performance. Meanwhile, we also test a variant of sGNN that is only based on powers of the A and has the same receptive field as GA-MLP-A (further details given in Appendix L). We see that its performance is comparable to that of GA-MLP-H. Thus, this demonstrates a scenario in which GA-MLP with common choices of Ω do not work well, but there exists some choice of Ω that is a priori unknown, with which GA-MLP can achieve good performance. In contrast, a GNN model does not need to rely on the knowledge of such an oracle set of operators, demonstrating its superior capability of learning.

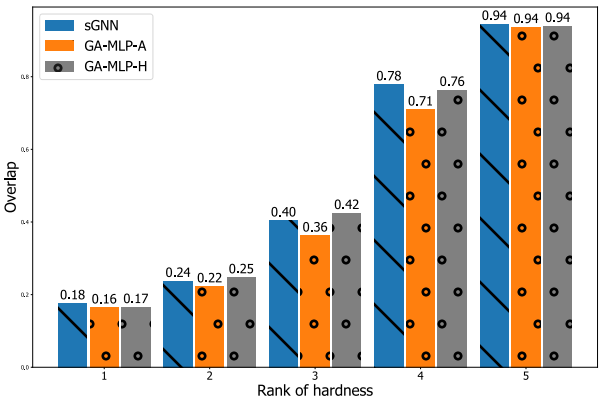

Figure 3: Community detection on binary SBM with 5 choices of in- and out-group connectivities, each yielding to a different SNR. Higher overlap means better performance.

7 Conclusions

We have studied the separation in terms of representation power between GNNs and a popular alternative that we coined GA-MLPs. This latter family is appealing due to its computational scalability and its conceptual simplicity, whereby the role of topology is reduced to creating 'augmented' node features then fed into a generic MLP. Our results show that while GA-MLPs can distinguish almost all non-isomorphic graphs, in terms of approximating node-level functions, there exists a gap growing exponentially-in-depth between GA-MLPs and GNNs in terms of the number of equivalence classes of nodes (or rooted graphs) they induce. Furthermore, we find a concrete class of functions that lie in this gap given by the counting of attributed walks. Moreover, through community detection, we demonstrate the lack of GA-MLP's ability to go beyond the fixed family of operators as compared to GNNs. In other words, GNNs possess an inherent ability to discover topological features through learnt diffusion operators, while GA-MLPs are limited to a fixed family of diffusions.

While we do not attempt to provide a decisive answer of whether GNNs or GA-MLPs should be preferred in practice, our theoretical framework and concrete examples help to understand their differences in expressive power and indicate the types of tasks in which a gap is more likely to

be seen – those exploiting stronger structures among nodes like counting attributed walks, or those involving the learning of graph operators. That said, our results are purely on the representation side, and disregard optimization considerations; integrating the possible optimization counterparts is an important direction of future improvement. Finally, another open question is to better understand the links between GA-MLPs and spectral methods, and how this can help learning diffusion operators.

ACKNOWLEDGEMENTS

We are grateful to Jiaxuan You for initiating the discussion on GA-MLP-type models, as well as Mufei Li, Minjie Wang, Xiang Song, Lingfan Yu, Michael M. Bronstein and Shunwang Gong for helpful conversations. This work is partially supported by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, NSF CHS-1901091, Capital One, Samsung Electronics, and the Institute for Advanced Study.

REFERENCES

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

László Babai, Paul Erdos, and Stanley M Selkow. Random graph isomorphism. *SIaM Journal on computing*, 9(3):628–635, 1980.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv preprint arXiv:2006.09252*, 2020.

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, 2017.

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

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. *arXiv preprint arXiv:2007.02133*, 2020a.

Ting Chen, Song Bian, and Yizhou Sun. Are powerful graph neural nets necessary? a dissection on graph classification. *arXiv preprint arXiv:1905.04579*, 2019a.

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

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Advances in Neural Information Processing Systems*, pp. 15868–15876, 2019c.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *arXiv preprint arXiv:2002.04025*, 2020b.

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

Daniel Flam-Shepherd, Tony Wu, Pascal Friederich, and Alan Aspuru-Guzik. Neural message passing on high order paths. *arXiv preprint arXiv:2002.10413*, 2020.

Vikas K. Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks, 2020.

Shunwang Gong, Mehdi Bahri, Michael M Bronstein, and Stefanos Zafeiriou. Geometrically principled connections in graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11415–11424, 2020.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Sergei Ivanov, Sergei Sviridov, and Evgeny Burnaev. Understanding isomorphism bias in graph data sets. *arXiv preprint arXiv:1910.12091*, 2019.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 7092–7101, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2018.

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

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.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9267–9276, 2019.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020a.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding–design provably more powerful gnns for structural representation learning. *arXiv preprint arXiv:2009.00142*, 2020b.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 3538–3545. Association for the Advancement of Artificial Intelligence, 2018.

Andreas Loukas. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l2bp4YwS.

Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *Advances in neural information processing systems*, pp. 10945–10955, 2019.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pp. 2153–2164, 6 2019a.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. volume 97 of *Proceedings of Machine Learning Research*, pp. 4363–4371, Long Beach, California, USA, 09–15 Jun 2019b. PMLR. URL http://proceedings.mlr.press/v97/maron19a.html.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Federico Monti, Karl Otness, and Michael M Bronstein. Motifnet: a motif-based graph convolutional network for directed graphs. In *2018 IEEE Data Science Workshop (DSW)*, pp. 225–228. IEEE, 2018.

Christopher Morris and Petra Mutzel. Towards a practical k-dimensional weisfeiler-leman algorithm. *arXiv preprint arXiv:1904.01543*, 2019.

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.

Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Victor M Preciado and Ali Jadbabaie. From local measurements to network spectral properties: Beyond degree distributions. In *49th IEEE Conference on Decision and Control (CDC)*, pp. 2686–2691. IEEE, 2010.

Emanuele Rossi, Fabrizio Frasca, Ben Chamberlain, Davide Eynard, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Alaa Saade, Florent Krzakala, and Lenka Zdeborová. Spectral clustering of graphs with the bethe hessian. In *Advances in Neural Information Processing Systems*, pp. 406–414, 2014.

Ryoma Sato. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *Advances in Neural Information Processing Systems*, pp. 4081–4090, 2019.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. *arXiv preprint arXiv:2002.03155*, 2020.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025. URL https://doi.org/10.1137/1116025.

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.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/wu19e.html.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5453–5462, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International Conference on Machine Learning*, pp. 7134–7143, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.