A New Perspective On the Expressive Equivalence Between Graph Convolution and Attention Models

Dai Shi 20195423@STUDENT.WESTERNSYDNEY.EDU.AU
Western Sydney University

Zhiqi Shao ZHIQI.SHAO@SYDNEY.EDU.AU
University of Sydney

Andi Han ANDI.HAN@SYDNEY.EDU.AU
University of Sydney

Yi Guo Y.GUO@WESTERNSYDNEY.EDU.AU
Western Sydney University

Junbin Gao JUNBIN.GAO@SYDNEY.EDU.AU
University of Sydney

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Graph neural networks (GNNs) have demonstrated impressive achievements in diverse graph tasks, and research on their expressive power has experienced significant growth in recent years. The well-known Weisfeiler and Lehman (WL) isomorphism test has been widely used to assess GNNs' ability to distinguish graph structures. However, despite being considered less expressive than other GNNs in graph-level tasks based on the WL test, two prominent GNN models, namely graph convolution networks (GCN) and attention-based graph networks (GAT), still exhibit strong performance in node-level classification tasks. In this paper, we present a comprehensive analysis of their expressive power using a novel evaluation metric: the number of linear regions. We demonstrate that by enhancing GCN with refined graph Ricci curvature, our proposed high-rank graph convolution network (HRGCN) can match or even surpass the prediction advantage of attention models. Thus, the two models exhibit equivalent node-level expressive powers. This fresh perspective highlights the evaluation of GNNs' expressive power in node-level classifications rather than solely at the graph level. Experimental results showcase that the proposed HRGCN model outperforms the state-of-the-art in various classification and prediction tasks.

Keywords: Graph Neural Networks; Expressive Equivalence; Number of Linear Regions

1. Introduction

Over the past decades, deep learning (DL) models have been developed as one of the most powerful tools in machine learning. It is well known that a feed-forward deep neural network is capable of approximating any Borel measurable function with a sufficient number of neurons (Hornik et al., 1989). Among enormous successful DL models, deep neural networks (DNN), convolutional neural networks (CNN), and recently the graph neural networks (GNN) were recognized as three milestones that have inspired countless developments based on them. Ever since these three classes of models were published, the discussions on how

© 2023 D. Shi, Z. Shao, A. Han, Y. Guo & J. Gao.

powerful they are in terms of their architecture, geometric feature, and bounds of learning capacity have never stopped. Among various metrics used to quantify the model learning capability, the number of linear regions (Lei et al., 2020; Montufar et al., 2014) reflects the model's expressive power of distinguishing the input data, separating and representing them into different affine spaces of the output domain. While in the last two decades, the estimate of the expressive power of DNN (Montufar et al., 2014; Pascanu et al., 2013) and CNN (Xiong et al., 2020) has been vigorously discussed, such results for GNNs are still on going. Thanks to the recent development on the expressive power of GNNs (Wijesinghe and Wang, 2022; Xu et al., 2018), the relationship between the number of linear regions of GNNs in terms of their model architectures is emerging. However, the comparison between various GNNs expressive power in terms of linear regions is still unclear. In addition, existing evaluation metrics of GNN's expressive power varies between literatures. Specifically, most of the literatures (Wijesinghe and Wang, 2022; Xu et al., 2018) investigate the expressive power of GNNs via graph level distinguishability (i.e., Weisfeiler and Lehman test of isomorphism). While these methods show significant improvements via GNNs graph level classification performance, the reason for those "less" powerful GNNs can still yield good performance on node level classification task is still unclear.

In this paper, we resolve the challenges illustrated above by evaluating two selected "less" powerful GNNs: graph convolution networks (GCN) (Kipf and Welling, 2016) and attention based graph learning models (Veličković et al., 2017) via a **new expressive power metric**, that is the **number of linear regions.** Despite been labelled as less powerful via graph level, we explicitly show how and why these two models perform/express well in the node classification tasks. Our comparison result shows that the reason for causing different expressive power between GCN and attention-based models is determined by whether the nodes features are considered or not. Specifically, the consideration of nodes features gives attention models higher capacity than GCN by enhancing the rank of the re-weighted adjacency matrix (see Section 4 for details). We further note that in the sequel, when we present expressive power of GNNs, we mean the model's ability of generating linear regions unless we further specify.

Furthermore, to potentially further enhance GCN such that it can produce identical or superior expressive power to attention based models, one scheme is to consider a re-weighting matrix that aggregates both graph topology and node feature information. In terms of graph topological information, we consider graph Ollivier Ricci curvature (Ollivier, 2007) which has shown to enhance the performance of graph neural networks (Li et al., 2022; Ye et al., 2019). To enable Ollivier Ricci curvature to incorporate nodes features, we refine the curvature with node feature distance while maintaining the initial properties of Ollivier Ricci curvature. We show the details on how this curvature is defined and its properties in Section 5. Moreover, we prove that the enhanced model is capable of producing the same number of linear regions and thus has identical expressive power compared to attention models. We also verify that the refined Ricci curvature enhanced GCN model can be understood as one-step graph Ricci flow and has the potential to alleviate over-squashing phenomenon and improve the prediction tasks on heterophilic graphs.

Contribution and Outline To our knowledge, this is the first study to compare graph learning models in terms of their expressive power which measured by the capability of

generating linear regions. In Section 2 we show a detailed literature review on the research development in the areas that are considered in this paper. In Section 3 we provide basic notions and quantify the number of linear regions based on graph learning model architectures. After that, in Section 4 we derive the expressive upper bound of attention based models and theoretically show that this upper bound is higher than its GCN counterpart. In Section 5 we define the refined graph Ricci curvature and prove that it can be considered as one of the enhancement schemes to let GCN potentially produce identical or even higher expressive power than attention based models. We then make a direct comparison on the expressive power differences between two evaluation metrics: number of linear regions and WL test of isomorphism. In addition, we also prove that the computation within the new curvature model is equivalent to the classic graph Ricci flow and has the potential to handle the over-squashing problem (Topping et al., 2021) in the original GCN model. Furthermore, we verified the random perturbation that we further introduced to the proposed model can not only help the model to generate higher expressive power than attention model but also prevent our model from the over-smoothing issue (Cai and Wang, 2020). Lastly, we test the proposed model on several real-world datasets and show its state-of-the-art experimental outcomes in Section 6.

2. Related Works

Expressive Power of DNN and GNN models The expressive power measured by the number of linear regions generated by deep learning models was firstly studied by Pascanu et al. (2013). The paper proves the number of linear regions is upper bounded by $\sum_{i=0}^{n_0}\binom{n_1}{i}$ for a one-layer fully connected ReLU neural network with n_0 inputs and n_1 neurons. Furthermore, a lower bound was also derived in (Pascanu et al., 2013) as $(\prod_{l=0}^{L-1}\{\frac{n_l}{n_0}\})\sum_{i=0}^{n_0}\binom{n_L}{i}$ for the maximum number of linear regions of a fully-connected ReLU network with n_0 inputs and L hidden layers of widths $n_1, ..., n_L$. The lower bound estimate was further improved by Montufar et al. (2014) as $(\prod_{l=0}^{L-1}\{\frac{n_l}{n_0}\})^{n_0}\sum_{i=0}^{n_0}\binom{n_L}{i}$. Generalized from their works, the discussion on the expressive power of GNNs has also conducted recently. The most famous one in this filed is proposed in Xu et al. (2018) in which the tool named as Weisfeiler and Lehman (WL) test of isomorphism is utilized to measure GNNs power of distinguishing different graph (structures). Wijesinghe and Wang (2022) further generalize this idea by developing a GNN model that is more expressive than the standard WL test via sub-graph hierarchy. Most recently, the lower and upper bound of the maximum linear regions of GCN is also developed in (Chen et al., 2022).

Attention Based Graph Learning Models The initial idea of attention based learning models was firstly established to help the models attend to the structural importance of the data (Mnih et al., 2014). After that, the mechanism was successfully adopted by models for various tasks including image classification (Mnih et al., 2014) and captioning (Xu et al., 2015), image question answering (Yang et al., 2016), natural language question answering (Kumar et al., 2016). More recently, there has been a growing interest in attention models for graphs. The graph attention mechanism was firstly developed in (Veličković et al., 2017) and extensively applied into many tasks both homogeneous (Lee et al., 2018b) and heterogeneous graphs (Lee et al., 2018a; Shang et al., 2018). Although the attention coefficients are generated based on slightly different mechanisms in these papers, the approaches share

Table 1: The rank of (re-weight) adjacency matrices of GCN and GAT. From Table 1 one can check that rank degeneracy phenomenon widely exists in all commonly analyzed benchmarks. In particular, even the repeated rows are removed, the adjacency matrix ($\hat{A}$) utilized in GCN is still with large number of rank degeneracy whereas the re-weighted adjacency matrices in GAT ($\theta \odot \hat{A}$) is with much larger number of ranks.

Datasets	**Cora**	**Citeseer**	**Pubmed**	**Computers**	**CS**	**Physics**	**Photo**
Number of Nodes	2708	3327	19717	13752	18333	34493	7650
Number of Repeated Rows	83	252	2902	35	115	81	15
Rank of $\hat{A}$	2401	2780	7596	13241	17146	33799	7501
Rank of $\theta \odot \hat{A}$	2638	3090	19604	13440	17817	33994	7473

the common ground in that the attention is imported to allow models to adapt and focus on the importance which is the task relevance of the data.

Graph Ricci Curvature Graph Ricci curvature is a discrete analogue of Ricci curvature on Riemannian manifolds, which is useful in identifying tumor-related genes in bioinformatics (Sandhu et al., 2015), predicting and managing the financial market risks (Sandhu et al., 2016) and detecting network backbone and congestion (Ni et al., 2015). One of the major paths to define graph curvature is through Ollivier's discretization in metric space (Ollivier, 2007). Ricci curvature is a type of edge-based curvature which captures the property (relative importance) of the underlying graph. Recently, graph Ricci curvature has been considered to enhance the capacity of GNNs. For example Ye et al. (2019); Li et al. (2022) applied Ollivier Ricci curvature to construct attention coefficients. Topping et al. (2021) defined a refined Forman curvature to adjust the over-squashing and bottleneck issues within the GNN learning process.

3. Preliminaries

Graphs and graph convolutional network In this section, we introduce some preliminaries on graphs, GCN, graph Ricci curvature and linear regions of neural networks. We denote a graph $\mathcal{G} = (V, E)$ where V, E represent the sets of vertices and edges, respectively. We also consider $X = [x_1^\top; \ldots; x_n^\top] \in \mathbb{R}^{n \times d_0}$ as the feature matrix of the n nodes with each node feature vector $x_i \in \mathbb{R}^{d_0}$. We also let $A \in \mathbb{R}^{n \times n}$ be the adjacency matrix of the graph $\mathcal{G}$ and $\hat{A} = D^{-1/2}(I + A)D^{-1/2} \in \mathbb{R}^{n \times n}$ be the symmetrically normalized adjacency matrix with D as the degree matrix of $I + A$. We recall the propagation of a GCN layer (Kipf and Welling, 2016) is given by

$$H^{(\ell+1)} = \sigma(\hat{A}H^{(\ell)}W^{(\ell)}), \quad H^{(0)} = X, \tag{1}$$

where $\sigma(\cdot)$ is an activation function and $W^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ is the weight matrix at layer ℓ.

Attention based graph networks Attention based graph networks contain one (or a set of) matrices (denoted as $\mathcal{T}$) whose entries are learnable attention coefficients that element-wisely multiply to the graph adjacency matrix, i.e., $\mathcal{T} \odot \hat{A}$. Without loss of generality, we

consider the general attention models in which the attention coefficients are generated from various attention mechanisms defined as functionals in feature space domain. Accordingly, the graph attention model at layer l is defined as:

$$H^{(\ell+1)} = \sigma(\theta \odot \hat{A}H^{(\ell)}W^{(\ell)}), \quad H^{(0)} = X, \tag{2}$$

where $\theta \in \mathbb{R}^{n\times n}$ is the matrix that contains the attention coefficients. Each entry of θ represents the attention from a central node to one of its peripheral nodes which is computed in its neighbourhoods. For example, in (Veličković et al., 2017), the attention coefficients are computed from softmax function that is: $\theta_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}$, where $\mathcal{N}_i$ stands for all first order neighbourhoods of point x_i, and $e_{ij} = a(w^T x_i, w^T x_j))$ is obtained from a function $a(\cdot)$ with $w \in \mathbb{R}^{d_0\times d'}$ as the trainable coefficient parameter for all nodes and their neighbourhoods. Furthermore, the row normalization in θ ensures $\sum_j \theta_{i,j} = 1$, $\forall i$.

Graph Ricci curvature. In this paper, we particularly focus on the graph Ollivier Ricci curvature defined in (Ollivier, 2007; Lin et al., 2011) on graph. Specifically, given two connected nodes, their Ricci curvature illustrates how difficult the mass (information) from one distribution generated from one node with its neighbours transact to another distribution defined from another node with its neighbours, compare to the flat case. Therefore, before we introduce the definition, we define a probability measure at node $i \in V$ as for a given $\alpha \in [0, 1]$

$$m_i(j) = \begin{cases} \alpha, & j = i \\ \frac{1-\alpha}{|\mathcal{N}_i|}, & j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}$$

where $|\mathcal{N}_i|$ is the size of $\mathcal{N}_i$, i.e. the degree of x_i. We highlight that this is the original definition in (Ollivier, 2007) and there exist many alternatives to define m_i as long as each m_i generates a discrete distribution over every node in V. Then the graph Ollivier Ricci curvature between node i, j is defined as

$$\kappa(i, j) = 1 - \frac{W_1(i, j)}{d_s(i, j)},$$

where $d_s(i, j)$ is the shortest path distance on $\mathcal{G}$ between nodes i, j and $W_1(i, j)$ is the L_1-Wasserstein distance computed as $W_1(i, j) = \inf_\Gamma \sum_{i'}\sum_{j'} \Gamma_{i'j'} d_s(i', j')$ where Γ is the joint distribution satisfies the coupling conditions, i.e., $\sum_{i'} \Gamma_{i'j'} = m_j(j')$, $\sum_{j'} \Gamma_{i'j'} = m_i(i')$ for all i', j'. Note that if node i and j are not connected, we set the edge weight as 0 as in (Ollivier, 2009). Similar to the settings from previous literatures (Li et al., 2022; Ni et al., 2019) we set curvature as 1 for nodes with self-loop.

Linear regions of GCNs. Here we consider a general form of GCN given by $H^{(\ell+1)} = \sigma(\mathcal{A}H^{(\ell)}W^{(\ell)})$ for some symmetric matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ with the same structure as the adjacency matrix where the only nonzero entries are on the edges. When $\mathcal{A} = \hat{A}$, this becomes the original GCN (Kipf and Welling, 2016). From this point on, we restrict the activation function σ in GCN to be the Rectifier Linear Unit (ReLU). In this case, GCN can be written as a piecewise linear function.

Definition 1 (Activation patterns and linear regions (Montufar et al., 2014)) *Let $\mathcal{F}$ be an L-layer GCN with k neurons in total. An activation pattern is a function of the k (pre-activation) neurons (denoted as $z_i(X), i = 1, ..., k$) where X is the input. An activation pattern of $\mathcal{F}$ is a function $\mathcal{P}$ from $\{z_i(X)\}$ to $\{-1, 1\}$. Let θ be a fixed set of parameters of $\mathcal{F}$. The region corresponding to $\mathcal{P}$ and θ is $\mathcal{R}(\mathcal{P}, \theta) = \{X : z_i(X) \cdot \mathcal{P}(z_i(X)) > 0, \forall\, i\}$. A linear region of $\mathcal{F}$ at θ is a non-empty set $\mathcal{R}(\mathcal{P}, \theta)$. Then the number of linear regions of $\mathcal{F}$ is $\mathcal{R}_{\mathcal{F},\theta} = \#\{\mathcal{R}(\mathcal{P}, \theta) : \mathcal{R}(\mathcal{P}, \theta) \neq \emptyset\}$, where for a set Q, $\#Q$ denotes the number of elements in Q.*

The following Lemma derives the number of linear regions of a single layer of GCN.

Lemma 1 (Number of linear regions of one-layer GCNs (Chen et al., 2022)) *Let $X \in \mathbb{R}^{n\times d_0}$ and $H^{(1)} = \sigma(\mathcal{A}XW) \in \mathbb{R}^{n\times d_1}$ be the input and output of a GCN $\mathcal{F}$. Let $\widetilde{\mathcal{A}}$ be the adjacency matrix that excludes the repeated rows, and $D^* = rank(\mathcal{A})$. Furthermore, assume that total number of p parameters are drawn from some distribution μ which has densities with respect to Lebesgue measure in $\mathbb{R}^p$. Then the number of linear regions of $\mathcal{F}$ is $\mathcal{R}_{\mathcal{F},\theta} = \left(\sum_{i=0}^{d_0} \binom{d_1}{i}\right)^{D^*}$ almost surely. Moreover, the expectation is $E_{\theta\sim\mu}(\mathcal{R}_{\mathcal{F},\theta}) = \left(\sum_{i=0}^{d_0} \binom{d_1}{i}\right)^{D^*}$.*

Based on Lemma 1, the number of linear regions (expressive power) of the GCNs depends on d_0, d_1 and D^*, for any two models that with fixed input and output feature dimension, the only variable that determines their expressive power is D^*. This observation provides a way to study the expressive power between GCN and graph attention based models, and a chance of enhancing GCN to achieve identical or even higher expressive power compared to graph attention models by preserving the rank of $\mathcal{A}$.

4. Expressive Power Comparison Between GCN and Attention Models

In this section, we show the difference in the number of linear regions between the original GCN model and attention based models. Compared to original GCN model defined in (1), in which only the graph connectivity information is considered, the attention based models defined in (2) aggregates both connectivity and feature information of the graph and offers a re-weighting process onto graph adjacency matrix. In terms of the graph adjacency information ($\hat{A}$) processed in GCN, however, there are many possibilities for rank degeneracy on $\hat{A}$. For example, if $\mathcal{G}$ contains a fully connected subset whose nodes may or may not connect to common nodes outside the subset, as a consequence, the row values of these nodes in $\hat{A}$ will be identical causing GCN failed to distinguish them. Table 1 summarizes the rank degeneracy phenomenon existed in GCN and GAT using adjacency matrices in citation networks. Please refer to Section 6 and Appendix A.1 for more detailed discussions. The next lemma shows that in real-world datasets, with the help of the attention coefficients, the chance of having a rank degeneracy re-weighted adjacency matrix is zero.

Lemma 2 *Let $S_1 := \{M \in \mathbb{R}^{n\times n} | m_{i,j} \geq 0, m_{i,j} = m_{j,i}, \sum_j m_{i,j} = 1 \forall i\}$ be the space that contains all normalized matrices of size $n \times n$, with symmetric and positive entries, and $S_2 \subset S_1, s.t.\ \forall M \in S_2, det(M) = 0$ be the subset of all matrices with rank degeneracy from S_1. Let μ be a measure defined on S_1, then we have $\mu(S_2) = 0$.*

Therefore S_1 can be regarded as the space that contains all re-weighted adjacency matrix (i.e., $\mathcal{T} \odot \hat{A}$), and S_2 as the space that contains all possible $\hat{A}$'s that degenerate. We note that the conclusion in Lemma 2 also holds when $\hat{A}$ has graph structure (i.e., if two nodes are disconnected, the edge weight of them remains 0 after the re-weighting process). The proof of this lemma relies on the fact that the manifold defined by S_1 is higher dimensional than the manifold S_2. Hence for any measure μ on S_1, we have $\mu(S_2) = 0$. We include the whole proof of this lemma in Appendix A as well. By direct comparison, it is clear to see that although it is possible to have a rank degenerated $\hat{A}$, it is almost impossible to have a rank degenerated $\mathcal{T} \odot \hat{A}$ from real-world datasets. Hence, let $\mathcal{R}_{\mathcal{F},\theta}(ATT)$ and $\mathcal{R}_{\mathcal{F},\theta}(GCN)$ be the number of linear regions generated from attention based and GCN model respectively, based on Lemma 1 and the Lemma 2 we showed above, if we fix the input and output feature dimensions as d_0 and d_1, we have:

$$\mathcal{R}_{\mathcal{F},\theta}(ATT) \geq \mathcal{R}_{\mathcal{F},\theta}(GCN) \tag{3}$$

We note that the equal sign appears only when the graph nodes are all with the same feature and connectivity (i.e., the complete graph, with all node features identical). Based on the observation in the inequality 3 it is natural to ask the following question: Is it possible to develop a meaningful re-weighting scheme to $\hat{A}$ other than attention mechanism such that the new model has the identical or even higher expressive power to attention models? In the next section, we will propose a new model and show this task can be done by a refined version of graph Ollivier Ricci curvature.

5. High Rank Graph Convolution Network (HRGCN)

To properly define the refined graph Ricci curvature, we recall the definition of Ricci curvature mentioned in Section 3, that is: $\kappa(i,j) = 1 - \frac{W_1(i,j)}{d_s(i,j)}$. Clearly, $\kappa(i,j)$ shows the potential of being a re-weighting coefficient since it illustrates the topological importance of neighboring nodes which plays an important role in the information aggregation (i.e., message-passing). In fact, as most of the existing GNN models are message-passing based (Gilmer et al., 2017), one can define a message-passing neural network (MPNN) to explicitly illustrate the importance of the inclusion of graph Ricci curvature as:

$$h_i^{(\ell+1)} = \phi_\ell \left(\bigoplus_{j \in \mathcal{N}_i} \psi_\ell(h_i^{(\ell)}, h_j^{(\ell)}) \right), \tag{4}$$

where $\mathcal{N}_i$ stands for the set of all neighbours of node i, ϕ_ℓ is the updated function, usually presented as the activation function, $\bigoplus$ is the aggregation function and ψ_ℓ is the message passing function which is usually trainable. It is not hard to check that whether a MPNN model is capable of delivering a high prediction outcome largely depends on how the feature information is aggregated via propagation. This observation directly shows the advantage of choosing Ricci curvature for model enhancement. However, $\kappa(i,j)$ cannot distinguish the nodes with the same connectivity and cannot escape rank degeneracy in $\hat{A}$, resulting a limited expressive power of the learning model. Therefore, to equip Ricci curvature with identical expressive power as the attention models, node feature information shall be considered. We define the refined graph Ricci curvature as follows:

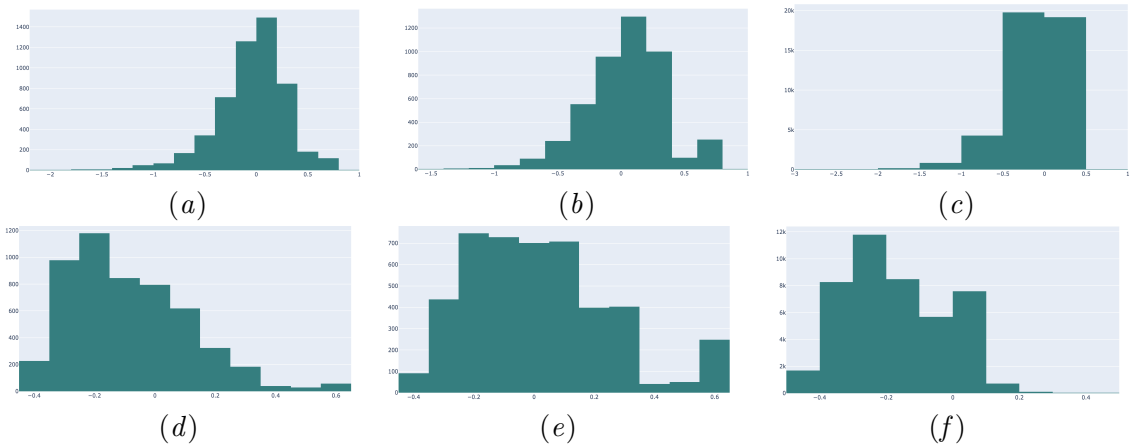

Figure 1: Changes of the graph Ricci curvature distribution before (first row) and after (second row) the computation conducted in HRGCN. The datasets on each row from left to right are **Cora, Citeseer** and **Pubmed**, we fixed $\alpha = 0.7$ as the initial mass for curvature computation. We observe HRGCN is able to smooth the curvature of graphs by contracting both the negative and positive curvatures.

Definition 2 (Refined Graph Ollivier Ricci Curvature) *The refined, feature information based graph Ollivier Ricci curvature is defined as:*

$$\widetilde{\kappa}(x_i, x_j) = \kappa(x_i, x_j) \times d(x_i, x_j), \tag{5}$$

where $d(x_i, x_j)$ is the Euclidean distance between two nodes' features.

The refined Ricci curvature maintains the sign of the original Ollivier Ricci curvature, and thus the topological information of the graph is preserved. In fact the sign of Ricci curvature is an important indicator for a list of graph related tasks and problems such as community detection (Ni et al., 2019) and over-squashing (Topping et al., 2021). Thanks to the following theorem, we can show that the adjacency re-weighting scheme induced from the refined Ricci curvature on GCN can balance or even surpass the prediction advantage in attention based models.

Theorem 1 (Expressive Equivalence) *Let D^*_{ATT} and D^*_{HRGCN} be the rank of $\theta \odot \hat{A}$ and $\eta \odot \hat{A}$, respectively, where θ is the matrix contains all learnable attention coefficients and η is the matrix with entries of the refined graph Ollivier Ricci curvature similarities that is:*

$$\eta_{ij} = Exp(-\widetilde{\kappa}_{ij})$$

Then we have $\mathcal{R}_{HRGCN} = \mathcal{R}_{ATT}$.

The proof of Theorem 1 is based on the fact that each row of the matrices of both $\theta \odot \hat{A}$ and $\eta \odot \hat{A}$ lost one degree of freedom and resulted in an identical dimension of the space that contains them. The loss of the degree of freedom is due to the feature of normalization

in the attention model and the definition of graph Ricci curvature in HRGCN since the diagonal of the matrix $\eta \odot \hat{A}$ is equal to $\text{Exp}(0) = 1$. We leave the detailed proof of Theorem 1 in Appendix A. Based on Theorem 1 the inclusion of the refined graph Ollivier Ricci curvature protects the rank of $\hat{A}$ from the rank degeneration similar to the attention matrix. Therefore, one can denote the computation within one single layer of HRGCN is:

$$H^{(\ell+1)} = \sigma(\eta \odot \hat{A} H^{(\ell)} W^{(\ell)}), \quad H^{(0)} = X.$$

Similar to the graph attention model in (Veličković et al., 2017), we set σ as Leaky_relu activation function.

Compared with WL Test of Isomorphism We briefly show the difference between two measures of expressive power of GNNs: number of linear regions and WL test of isomorphism. Specifically, the WL test acts on the connected node features (known as multisets) via subtree graph structure and a GNN can only reach as powerful as WL test in terms of distinguishing non-isomorphic graphs (Xu et al., 2018). Therefore it is not hard to check that WL test is unable to illustrate how expressive one GNN is in terms of node level learning tasks. Our proposed method, however, aims to quantify what the maximal power of a GNN model is in terms of the capability in distinguishing node features on *a single graph*.

Relationship with Graph Ricci Flow Graph Ricci flow (Weber et al., 2016) is a discrete version of Ricci flow on Riemannian manifold (Hamilton, 1982), which iteratively shrinks the positive edges and pushes away the negative edges. Here we demonstrate the connection of the proposed HRGCN with graph Ricci flow as follows.

Let $a_{i,j}$ represent the weight between nodes i and j. The Ricci flow on graph (Ni et al., 2019) updates the weights iteratively by $a^{+}_{i,j} = d_s(i,j)(1-\kappa_{i,j})$, where $d_s(i,j)$ is the shortest path distance between node i and j and $\kappa_{i,j}$ is the Olivier Ricci curvature for the edge i and j, both calculated using the weight $a_{i,j}$ at current iteration. For unweighted graph, if there exists an edge between i, j, then at first iteration $d_s(i,j) = a_{i,j} = 1$ and thus the process can be interpreted as increasing the edge weight for negatively curved edges and decreasing the edge weight for positively curved ones. It is easy to verify that the curvature re-weighting process, i.e., $\eta \odot \hat{A}, \eta = \text{Exp}(-\kappa_{i,j} d(x_i, x_j))$ of HRGCN aligns with the property of graph Ricci flow as $\eta_{i,j} > 1$ for $\kappa_{i,j} < 0$ and $\eta_{i,j} < 1$ for $\kappa_{i,j} > 0$. Thus the proposed re-weighting scheme smooths the curvature via the re-weighted matrix $\eta \odot \hat{A}$. Fig. 1 shows this phenomenon for citation networks. It is clear that the computation in HRGCN shrinks both positive Ricci curvatures and the negative curvatures to a narrower range compared to the curvature based on the initial weights from the adjacency matrix. This has the potential of alleviating the problem of bottleneck which will be discussed.

Over-Squashing and Bottleneck Based on the relationship with Ricci flow, HRGCN allocates larger edge weight to the edge that initially with negative curvatures. From the perspective of graph neural networks, a larger edge weight corresponds to strong connection between the nodes. From (Topping et al., 2021), we see that negative edges are responsible for the over-squashing and bottleneck in GNNs where the long-range dependencies of the nodes cannot be captured. In (Topping et al., 2021), a remedy is proposed by adding edges in the neighbourhood leading to negatively curved edges. Here we show the proposed

Table 2: Sensitivity Bottleneck value comparison between GCN and HRGCN in both homophily and heterophily networks. We can see HRGCN produces stronger connectivity to the negative curvature edges and has lower bottleneck values to prevent model from over-squashing.

Datasets	**Cora**	**Citeseer**	**Pubmed**	**Cornell**	**Wisconsin**	**Actor**
Minimum Curvature	-0.539	-0.516	-0.575	-0.155	-0.159	-1.60
Sensitivity (GCN)	0.0006	0.051	0.0003	0.011	0.026	0.0008
Sensitivity (HRGCN)	0.024	0.094	0.0012	0.031	0.029	0.0054
Bottleneck Value (GCN)	6901.4	6099.8	63352.7	130.6	161.86	11813.5
Bottleneck Value (HRGCN)	5985.4	4924.2	50610.8	121.2	130.32	9123.4

$\eta_{ij} = \text{Exp}(-\tilde{\kappa}_{ij})$ can also alleviate the issue by increasing the edge weight for negatively curved edges. The following Lemma quantifies the sensitivity of propagation in the form $H^{(\ell+1)} = \sigma_\ell(\mathcal{A}H^{(\ell)}W_\ell)$. This Lemma adapts Lemma 1 in (Topping et al., 2021).

Lemma 3 *Consider the propagation $H^{(\ell+1)} = \sigma_\ell(\mathcal{A}H^{(\ell)}W_\ell)$ at layer ℓ with $H^{(0)} = X$ and $\mathcal{A} = \Lambda \odot \hat{A}$ for some $\Lambda \in \mathbb{R}_+^{n\times n}$. Let $h_v^{(\ell)}$ represents the feature of node v at layer ℓ. Suppose $|\sigma'_\ell| \leq \alpha$ and $\|W_\ell\|_2 \leq \beta$ for all ℓ. Then we have for any node u, v with $d_{\mathcal{G}}(u, v) = \ell + 1$, we have $\left\|\frac{\partial h_v^{(\ell+1)}}{\partial x_u}\right\|_2 \leq (\alpha\beta)^{\ell+1}(\mathcal{A}^{\ell+1})_{uv}$.*

Lemma 3 shows when the derivative of activation functions and the weights are bounded, the sensitivity of the features on the input depends critically on the matrix $\mathcal{A}$. We show the details of the proof in Appendix A. Since negative curvatures are responsible for the bottleneck problem (Topping et al., 2021) and in HRGCN and a negative curvature will give a larger weight (strong connectivity) due to $\eta_{i,j} = \text{Exp}(-\widetilde{\kappa}_{i,j})$, thus HRGCN naturally has the potential of preventing the sensitivity of the node feature respect to the input from diluting away which happens in GCN.

Another measurement on the bottleneck problem is through the notion of *Betweenness Centrality* (Freeman, 1977) which illustrates the frequency of a node that appears in the minimal path of distinct pairs of nodes, that is: $c_B(u) = \sum_{s,t\in V} \frac{\sigma(s,t|u)}{\sigma(s,t)}$, where $\sigma(s, t)$ is the number of shortest (s, t)-path and $\sigma(s, t|u)$ is the number of shortest paths between s and t that route through node u. According to (Topping et al., 2021), the bottleneck value of the graph is defined as:

$$b_{\mathcal{G}} = \frac{1}{n}\sum_i c_B(i). \tag{6}$$

When graph $\mathcal{G}$ is complete, $b_{\mathcal{G}} = 1$. Thus $b_{\mathcal{G}}$ shows how far away a given graph $\mathcal{G}$'s topology is from the complete graph in which any pair of nodes are connected, and thus no bottleneck occurs. In (Topping et al., 2021), this was the motivation of conducting the graph-rewiring scheme to fix the bottleneck problem. In Table 2, we measure the bottleneck problem via both sensitivity and bottleneck value to demonstrate the effectiveness of HRGCN in handling the bottleneck problem. For the sensitivity comparison, we select the node u as one of the nodes with its edge that contains the most negative curvature and select the node v which is one of the 2-hop neighbours (as models are set as two layers by default) of u with

Table 3: Test Accuracy scores(%) for HRGCN in six heterophily graph benchmarks. Accuracies are highlighted in **bold** when HRGCN outperforms GAT and GCN.

Methods	Cornell	Wisconsin	Texas	Actor	Chameleon	Squirrel
MLP-2	91.30±0.70	93.87±3.33	92.26±0.71	38.58±0.25	46.72±0.46	31.28±0.27
GAT	76.00±1.01	71.01±4.66	78.87±0.86	35.98±0.23	63.90±0.46	42.72±0.33
APPNP	91.80±0.63	92.00±3.59	91.18±0.70	38.86±0.24	51.91±0.56	34.77±0.34
H2GCN	86.23±4.71	87.50±1.77	85.90±3.53	38.85±1.77	52.30±0.48	30.39±1.22
GCN	66.56±13.82	66.72±1.37	75.66±0.96	30.59±0.23	60.96±0.78	45.66±0.39
Mixhp	60.33±28.53	77.25±7.80	76.39±7.66	33.13±2.40	36.28±10.22	24.55±2.60
GraphSAGE	71.41±1.24	64.85±5.14	79.03±1.20	36.37±0.21	62.15±0.42	41.26±0.26
HRGCN	**78.25±0.25**	**91.01±1.55**	**82.25±0.91**	**37.21±0.29**	56.81±0.12	44.28±0.91

the middle node v' such that the edge $e_{v,v'}$ has the second smallest curvature within all edges of v. Therefore, a larger sensitivity value illustrates a stronger preservation of the model in terms of long range dependencies. We fixed $\alpha = 0.7$ for all curvature computations.

Further Improvement from Random Perturbation It is possible to observe that two nodes have the same connectivity and features in the real-world datasets. In this case, both the feature-based attention model and HRGCN fail to distinguish these nodes as the Euclidean distance between nodes goes to 0, causing rank degeneracy for the re-weighting matrix. In this paper, we address this problem by inserting a random perturbation $\epsilon \sim U(0, 0.01)/1000$ s.t.$\epsilon < \min(\eta_{i,j} \odot \hat{a}_{i,j})$ [1] to the non-zero entries of $\eta \odot \hat{A}$ to ensure the model's distinguishability. Moreover, we show this operation is capable of lifting and stabilizing system's Dirichlet energy and thus has the advantage of preventing the model from over-smoothing. We show our conclusion as the theorem 3 in Appendix A.2.1

6. Experiment

In this section, we show a variety of numerical tests to solidify our theoretical analysis. Section 6.1 tests the performance of HRGCN on seven citation benchmarks. Section 6.2 shows that with greater expressive power compare to GCN, our proposed model can even handle the node classification task in heterophily graph datasets. Section 6.3 presents the ablation study to show HRGCN is robust to the changes of the model parameters. Moreover, we show the performance of HRGCN in graph level classification (pooling) in Appendix A.3. All experiments were conducted using PyTorch on NVIDIA® Tesla V100 GPU with 5,120 CUDA cores and 16GB HBM2 mounted on an HPC cluster.

6.1. Node Classification for HRGCN

Datasets and Setup We tested HRGCN model against the state-of-the-arts on seven node classification datasets. The task for node classification is conducted on several benchmark citation networks: **Cora, Citeseer, Coauthor CS and Physics**. In terms of the model setup, HRGCN is designed with two curvature assisted convolutional layers to compute graph embedding. The hidden layer output is followed by softmax activation function

1. $U(0, 0.01)$ stands for an uniform distribution with low and upper bound as 0 and 0.01 respectively.

Table 4: Test Accuracy for citation networks with standard deviation after ±. The top results are highlighted in **First**, **Second** and **Third**.

Method	Cora	Citeseer	PubMed	CS	Physics	Computers	Photo
MLP	55.1	59.1	71.4	88.3±0.7	88.9 ± 1.1	44.9±0.8	69.6±3.8
MoNet	81.7	71.2	78.6	90.8±0.6	92.5±0.9	83.4±2.2	91.2±1.3
GS-mean	79.2	71.2	77.4	**91.3±2.8**	**93.0±0.8**	82.4±0.8	**91.4±1.3**
GCN	81.5±0.5	70.9±0.5	79.0±0.3	91.1±0.5	92.8±1.0	**82.6±2.5**	91.2±1.2
GAT	**83.0±0.7**	**72.5±0.7**	**79.0±0.3**	90.5±0.6	92.5±0.9	78.0±1.9	85.1±2.3
GIN	81.0±1.1	70.5±0.9	78.3±1.2	91.2 ±1.4	88.5±0.6	77.1±0.4	84.8±0.7
CurvGN	**82.6±0.6**	**71.5±0.8**	**78.8±0.6**	**92.9±0.4**	**94.3±0.2**	**86.5±0.7**	**92.5±0.5**
HRGCN	**83.6±0.4**	**71.8±0.3**	**80.1±0.2**	**93.4±0.6**	**95.9±0.5**	**87.4±0.3**	**92.9±0.1**

for the final prediction. Most of hyperparameters were set the default values except from learning rate, weight decay, hidden units, dropout ratio, negative slop of leaky_relu function. We used grid search to tune the hyperparameters. The hyperparameter values and tuning results are listed in Appendix A.3. In addition, similar to (Ye et al., 2019), in Appendix A.3, we show that the computational cost of Ricci curvature can be relaxed by using approximation and parallel computation even in large datasets. We set the maximum number of epochs of 200 for all citation networks. All the datasets included in this series of experiment are split followed by the standard public processing rules. All the average test accuracy and standard deviations are summarized from 10 random trials.

Baseline The learning accuracy of HRGCN is compared against other methods. We consider multiple baselines that are applicable to the tasks. The test accuracy of the baseline models are retrieved from the published results: MLP, MoNet (Monti et al., 2017), WSCN, GINXu et al. (2018), (Morris et al., 2019), GraphSAGE with mean aggregation(GS-mean) (Hamilton et al., 2017), GCN (Kipf and Welling, 2016), GAT (Veličković et al., 2017) and Curvature graph networks (CurvGN) (Ye et al., 2019). The datasets for all baseline models are also split based on the standard public rules.

Results The top-3 test accuracy scores (in percentage) are highlighted in Table 4. HRGCN achieved highest predictive accuracy among all citation networks compared to baseline models. Both GAT and HRGCN models show superior prediction power within those relatively small datasets (i.e., **Cora, Citeseer** and **Pubmed**); whereas HRGCN remains producing the top accuracy in larger graph inputs.

6.2. Node Classification on Heterophily Graph datasets

In this section, we show that with the enhancement power from refined graph Ricci curvature, HRGCN can even handle the (heterophily) graph datasets in which the labels of nodes' neighbours are largely different compared to the (homophily) citation networks.

Datasets and Baselines We compare the learning outcomes of HRGCN to various baseline models,MLP with 2 layers (MLP-2), GCN,GAT,APPNP (Chien et al., 2020), H2GCN (Zhu et al., 2020), MixHop (Abu-El-Haija et al., 2019) and GraphSAGE (Hamilton et al., 2017). We test these models 10 times on **Cornell, Wisconsin, Texas, Film, Chameleon**

and Squirrel following the same early stopping strategy, and the same random data splitting method applied to the citation networks.

Results The testing accuracy and standard deviations of HRGCN for heterophily graph datasets are listed in Table 3. It is clear to see that HRGCN outperforms attention model (i.e. GAT) and GCN in most of datasets.

Discussion on Graph Density and Rank Degeneracy As we have illustrated previously, the rank degeneracy phenomenon is closely related to graph connectivity. Specifically, we observed that a degenerated graph has a high degree of symmetry. Apart from the example we have given in Section 4, if a graph is a complete graph, where every node is connected to every other node. Then the adjacency matrix is a matrix with all its entries equal to 1, and its rank is 1. Therefore, in general, any graph structure where the edges between nodes are highly symmetric, or where there is a high degree of homogeneity in the connections between nodes, will result in a normalized adjacency matrix with some rank degeneracy. Thus, HRGCN will tend to deliver a better performance if the graph becomes denser. This claim is supported by our experimental outcomes (i.e., Table 4) in which HRGCN produces much higher learning accuracy in denser graphs (i.e., **PubMed** and **Physics**).

6.3. Ablation Study

In this section, we conducted ablation studies on how the changes of the initial mass α affect the prediction accuracy of our model. The quantity of α is selected from $0.3, 0.6, 0.9$. We note that when α is small, meaning that model will take more information from the node neighbours rather than the node itself. In addition, we also modified the metric of computing the feature distances from Euclidean distance to both spherical and hyperbolic distances to see whether HRGCN is robust to the metric changes. Accordingly, we named two ablation models as HRGCN-S and HRGCN-H. We select **Cora, Citeseer** and **Pubmed** datasets and the results are contained in Table. 5. From the results, one can check that HRGCN is robust to the changes of the newly introduced model parameters. In addition, despite changes are applied, all HRGCN variants kept outperforming the baseline results in Table. 4, suggesting the effectiveness of incorporating the refined Ricci curvature enhance the expressive power of GNNs.

Table 5: Results of Ablation study by changing the quantity of α and the metric of computing feature distances. The highest accuracy is highlighted in **red**.

Methods	Cora	Citeseer	Pubmed
HRGCN$^{0.3}$	82.5±0.3	70.9±0.2	78.8±0.3
HRGCN$^{0.6}$	83.1±0.2	71.3±0.7	79.2±0.6
HRGCN$^{0.9}$	83.4±0.4	71.4±0.5	79.6±0.3
HRGCN-S	83.5±0.2	70.8±0.3	79.9±0.7
HRGCN-H	83.2±0.1	**72.0±0.6**	80.0±0.5
HRGCN	**83.6±0.4**	71.8±0.3	**80.1±0.2**

7. Final Remark and Conclusion

This paper introduced a new evaluation metric on measuring the expressive power of GNNs, that is the number of linear regions. We applied this new metric to compared the expressive differences between the original GCN and attention based models. We theoretically proved

that the advantage in attention based models can be matched and even surpassed by introducing a curvature re-weighting scheme to GCN which gave rise to our HRGCN model. This claim was verified by extensive numeric experiments where our proposed model outperformed baselines in various node-level and graph-level learning tasks. The positive results show the great potential and encourage us to explore it further. Our future research will focus on exploring the curvature guided graph surgery techniques such as graph re-wiring.

References

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Hao Chen, Yu Guang Wang, and Huan Xiong. Lower and upper bounds for numbers of linear regions of graph convolutional networks. *arXiv preprint arXiv:2206.00228*, 2022.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Richard S Hamilton. Three-manifolds with positive ricci curvature. *Journal of Differential geometry*, 17(2):255–306, 1982.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

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

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016.

John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. page 1666–1674, 2018a. doi: 10.1145/3219819.3219980. URL https://doi.org/10.1145/3219819.3219980.

John Boaz Lee, Ryan A. Rossi, Sungchul Kim, Nesreen K. Ahmed, and Eunyee Koh. Attention models in graphs: A survey. *CoRR*, abs/1807.07984, 2018b. URL http://arxiv.org/abs/1807.07984.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *International conference on machine learning*, pages 3734–3743. PMLR, 2019.

Na Lei, Dongsheng An, Yang Guo, Kehua Su, Shixia Liu, Zhongxuan Luo, Shing-Tung Yau, and Xianfeng Gu. A geometric understanding of deep learning. *Engineering*, 6(3): 361–374, 2020.

Haifeng Li, Jun Cao, Jiawei Zhu, Yu Liu, Qing Zhu, and Guohua Wu. Curvature graph neural network. *Information Sciences*, 592:50–66, 2022.

Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and koray kavukcuoglu. Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/09c6c3783b4a70054da74f2538ed47c6-Paper.pdf.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. Ricci curvature of the internet topology. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 2758–2766. IEEE, 2015.

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. Community detection on networks with ricci flow. *Scientific reports*, 9(1):1–12, 2019.

Yann Ollivier. Ricci curvature of metric spaces. *Comptes Rendus Mathematique*, 345(11): 643–646, 2007.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Romeil Sandhu, Tryphon Georgiou, Ed Reznik, Liangjia Zhu, Ivan Kolesov, Yasin Senbabaoglu, and Allen Tannenbaum. Graph curvature for differentiating cancer networks. *Scientific reports*, 5(1):1–13, 2015.

Romeil S Sandhu, Tryphon T Georgiou, and Allen R Tannenbaum. Ricci curvature: An economic indicator for market fragility and systemic risk. *Science advances*, 2(5):e1501495, 2016.

Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. Edge attention-based multi-relational graph convolutional networks. *arXiv preprint arXiv: 1802.04944*, 2018.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*, 2021.

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

Melanie Weber, Jürgen Jost, and Emil Saucan. Forman-ricci flow for change detection in large dynamic data sets. *Axioms*, 5(4):26, 2016.

Asiri Wijesinghe and Qing Wang. A new perspective on" how graph neural networks go beyond weisfeiler-lehman?". In *International Conference on Learning Representations*, 2022.

Huan Xiong, Lei Huang, Mengyang Yu, Li Liu, Fan Zhu, and Ling Shao. On the number of linear regions of convolutional neural networks. In *International Conference on Machine Learning*, pages 10514–10523. PMLR, 2020.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

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

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

Ze Ye, Kin Sum Liu, Tengfei Ma, Jie Gao, and Chao Chen. Curvature graph network. In *International Conference on Learning Representations*, 2019.