

ON THE EXPRESSIVE EQUIVALENCE BETWEEN GRAPH CONVOLUTION AND ATTENTION MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) have achieved remarkable successes in various graph tasks, and recent years have witnessed a flourishing growth in research regarding GNNs’ expressive power. The number of linear regions generated from GNNs is a recently considered metric that quantifies GNNs’ capacity. The estimate of the number of linear regions has been previously developed for deep and convolution neural networks (DNN and CNN). In this paper, we compare the expressive power of the classic graph convolution network (GCN) and attention based models in terms of their capability to generate linear regions. We show that the prediction advantage of attention models can be matched or even surpassed by enhancing GCN with refined graph Ricci curvature resulting the so-called high rank graph convolution network (HRGCN). Thus, the two models are equivalent to each other in terms of expressive power. Experimental results show that the proposed HRGCN model outperforms the state-of-the-art results in various classification and prediction tasks.

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 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 (Montufar et al., 2014; Lei et al., 2020) 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 (Chen et al., 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 this paper, we compare the expressive power between graph convolution networks (GCN) (Kipf & Welling, 2016) and attention based graph learning models. 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 matrix produced from the element-wise product between attention matrix and graph adjacency matrix.

Therefore, 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 (Ye et al., 2019; Li et al., 2022). 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 This paper, to our knowledge, is the first study to compare graph learning models in terms of their expressive power, measured by the capability of generating linear regions. In Section 2 we show a detailed literature review on the research development on 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. Moreover, 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 & Wang, 2020) within both GCN and attention models. Finally, 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 Deep Learning 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, \dots, n_L . The lower bound estimate was further improved by Montufar et al. (2014) as $(\prod_{l=0}^{L-1} \{\frac{n_l}{n_0}\})^{n_0} \sum_{i=0}^{n_0} \binom{n_L}{i}$. Followed by their works, various results on the lower and upper bounds for the maximal number of linear regions of fully-connected ReLU neural networks have been obtained Bianchini & Scarselli (2014); Telgarsky (2015); Serra et al. (2018); Hanin & Rolnick (2019). Furthermore, the number of linear regions of convolutional neural networks (CNN) has been explored recently in (Xiong et al., 2020) and the results have been extended to Maxout activation, a generalization of ReLU activation in (Montufar et al., 2021).

Expressive Power of Graph Neural Networks Discussions on the expressive power of graph neural networks (GNNs) have been established in various aspects. For example, Balcilar et al. (2021) analyzed the power of GNNs from the graph spectral perspective, and Xu et al. (2018) further explored the geometrical preservation (graph isomorphism) property of GNNs under the scope of Weisfeiler-Lehman test. Moreover, Bodnar et al. (2021) is the first who analyzes GNNs via the number of linear regions in which the aggregation function is linear and invertible. 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), machine translation (Bahdanau et al., 2014) and 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; Abu-El-Haija et al., 2017; Lee et al.,

2018a; Ryu et al., 2018) 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 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 and Graph neural networks 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). However, there exist various different notions of graph Ricci curvature. One definition is through Forman’s discretization defined on the polyhedral or CW complexes (Forman, 2003), which has been applied in (Das Gupta et al., 2018; DasGupta et al., 2020; Fournier et al., 2015). Another path to define graph curvature is through Ollivier’s discretization in metric space (Ollivier, 2007). Both Ollivier and Forman Ricci curvatures are edge-based curvature but with different ways of capturing the property 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 $X = [x_1^\top; \dots; x_n^\top] \in \mathbb{R}^{n \times d_0}$ 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 & Welling (2016) is given by

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

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. Initially published in Veličković et al. (2017) and further developed in Wang et al. (2019b); Schlemper et al. (2019); Wang et al. (2019a) on various fields, the graph attention model at layer l is defined as:

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

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. Ricci curvature is a scalar-valued curvature measuring the spread of geodesics on a Riemannian manifold. We particularly focus on the Ollivier Ricci curvature (Ollivier, 2007; Lin et al., 2011; Ni et al., 2019) on graph, which is an edge-based curvature. 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' .

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 & 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; Xiong et al., 2020; Chen et al., 2022)). *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 $z_i, i = 1, \dots, k$). An activation pattern of \mathcal{F} is a function \mathcal{P} from $\{z_i\}$ to $\{-1, 1\}$. Let θ be a fixed set of parameters of \mathcal{F} . The region corresponding to \mathcal{P} and θ is $\mathcal{R}(\mathcal{P}, \theta) = \{X : z_i \cdot \mathcal{P}(z_i) > 0, \forall i\}$. 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 \hat{A} be the adjacency matrix that excludes the repeated rows, and $D^* = \text{rank}(\hat{A})$. 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} (Please refer to Appendix A.1 for more details). 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. The next lemma shows that in real-world data sets, where nodes features can be assumed to be generated from some distributions, with the help of the attention coefficients, the chance of having a rank degeneracy re-weighted matrix is 0.

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 $\mu(S_2) = 0$.*

The proof of this lemma relies on the fact that the manifold \mathcal{M}_1 defined by S_1 is with one higher dimension than the manifold \mathcal{M}_2 , which is a submanifold of S_1 , defined by 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} based on proposition 1, it is almost impossible to have a rank degenerated $\mathcal{T} \odot \hat{A}$ from real-world data sets. 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:

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

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. 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:

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

$$\tilde{\kappa}(x_i, x_j) = \kappa(x_i, x_j) \times d(x_i, x_j) \quad (4)$$

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. This is because in the perspective of graph community detection (Ni et al., 2019), when two nodes are from the same community, the Ricci curvature on their edge is positive. In contrast, the Ricci curvature is negative when two nodes are from different communities. 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 curvatures that is:*

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

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

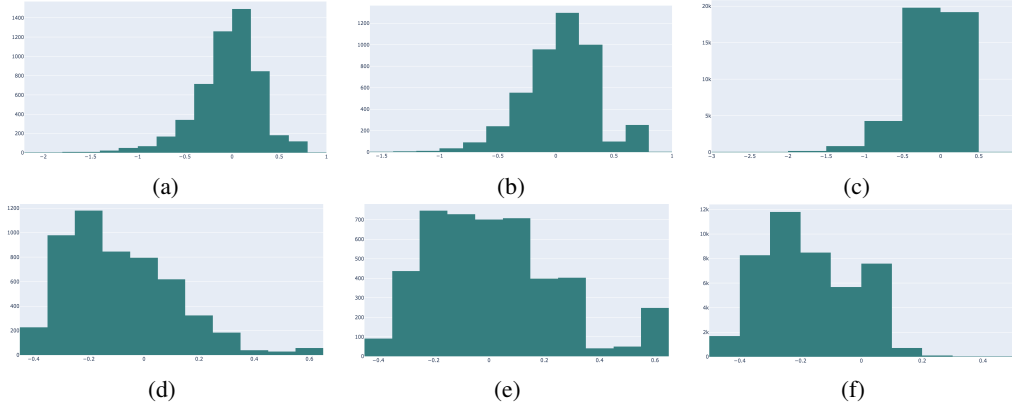


Figure 1: Changes of the graph Ricci curvature distribution before (first row) and after (second row) the computation conducted in HRGCN. The datasets on each row from left to right are **Cora**, **Citeseer** and **Pubmed**, we fixed $\alpha = 0.7$ as the initial mass for curvature computation. We observe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

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 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 produce the same preservation power to the rank of \hat{A} compared to the attention coefficients included in attention based models. Similar to the original GCN model, let HRGCN contain two curvature assisted convolution layers. 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.

Relationship with Graph Ricci Flow Graph Ricci flow (Weber et al., 2016; Ni et al., 2018; 2019) is a discrete version of Ricci flow on Riemannian manifold (Hamilton, 1982), which iteratively shrinks the positive edges and pushes away the negative edges. Graph Ricci flow has found applications for community detection (Ni et al., 2019), network alignment (Ni et al., 2018) and change detection on dynamic graphs (Weber et al., 2016). Here we demonstrate the connection of the proposed HRGCN with graph Ricci flow as follows.

Let $a_{i,j}$ represents the weight between nodes i and j . The Ricci flow on graph (Ni et al., 2019) for community detection 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 Ollivier 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 $\eta_{ij} = \text{Exp}(-\tilde{\kappa}_{ij})$ 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 $d_G(u, v) = \ell + 1$, 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 $\eta_{i,j} = \text{Exp}(-\tilde{\kappa}_{i,j})$, 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:

$$b_G = \frac{1}{n} \sum_i c_B(i). \quad (5)$$

When graph \mathcal{G} is complete, $b_G = 1$. Thus b_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 1, 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 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 data sets. 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. $\epsilon < \min(\eta_{i,j} \odot \hat{a}_{i,j})$ 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 present our conclusion as the theorem 2 below, and leave the proof in Appendix A.2.1.

Theorem 2 (Dirichlet Energy Preservation). *Let $\tilde{A} = \eta \odot \hat{A}$ and $\tilde{A}_\epsilon = \eta \odot \hat{A} - \epsilon$ be the re-weighted matrices of the curvature matrix η and the perturbed curvature matrix η_ϵ , respectively. Let $\tilde{\Delta}$ and $\tilde{\Delta}_\epsilon$ be the Laplacian matrices induced from \tilde{A} and \tilde{A}_ϵ , respectively. Then for any $\epsilon > 0$ and $\epsilon < \min(\eta_{i,j} \odot \hat{a}_{i,j})$, at any specific layer (i.e., l -th layer), we have:*

$$E_\eta(X^{(l)}) < E_{\eta_\epsilon}(X^{(l)})_\epsilon$$

Where $E_\eta(X^{(l)})$ and $E_\eta(X^{(l)})_\epsilon$ are the Dirichlet energy at layer k induced from η with and without perturbation ϵ .

Table 1: Sensitivity and Bottleneck value comparison between GCN and HRGCN in both homophily and heterophily networks. It is clear that HRGCN produces stronger connectivity to the negative curvature edges and has lower bottleneck values to prevent the 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

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 data sets. Moreover, we show the performance of HRGCN in terms of 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

Data sets We tested HRGCN model against the state-of-the-arts on seven node classification data sets. The task for node classification is conducted on several benchmark citation networks. The graph data sets **Cora**, **Citeseer** are relatively small and sparse with average node degree below 2. Other data sets, **Coauthor CS and Physics** are the co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge and **Amazon Photos and Computers** are the segments of the Amazon co-purchase graph in (McAuley et al., 2015). These data sets together with **PubMed** are denser and larger than **Cora** and **Citeseer** since they contains more than 10 thousands nodes and 20 thousands edges and with average nodes degrees more than 20.

Set-up HRGCN is designed with two curvature assisted convolutional layers to compute graph embedding. The hidden layer output is followed by softmax activation function 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 and curvature initial weight index α in training. 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 data sets. We set the maximum number of epochs of 200 for all citation networks. All the data sets 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 (Morris et al., 2019), GraphSAGE with mean aggregation(GS-mean) (Hamilton et al., 2017), GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017) and Curvature graph networks (CurvGN) (Ye et al., 2019). The data sets for all baseline models are also split based on the standard public rules.

Table 2: Test Accuracy (in percentage) for citation networks with standard deviation after \pm . 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 \pm 0.7	88.9 \pm 1.1	44.9 \pm 0.8	69.6 \pm 3.8
MoNet	81.7	71.2	78.6	90.8 \pm 0.6	92.5 \pm 0.9	83.4 \pm 2.2	91.2 \pm 1.3
GS-mean	79.2	71.2	77.4	91.3\pm2.8	93.0\pm0.8	82.4 \pm 0.8	91.4\pm1.3
GCN	81.5 \pm 0.5	70.9 \pm 0.5	79.0 \pm 0.3	91.1 \pm 0.5	92.8 \pm 1.0	82.6\pm2.5	91.2 \pm 1.2
GAT	83.0 \pm0.7	72.5\pm0.7	79.0\pm0.3	90.5 \pm 0.6	92.5 \pm 0.9	78.0 \pm 1.9	85.1 \pm 2.3
CurvGN	82.6\pm0.6	71.5\pm0.8	78.8\pm0.6	92.9\pm0.4	94.3\pm0.2	86.5\pm0.7	92.5\pm0.5
HRGCN	83.2\pm0.4	71.8\pm0.3	79.6\pm0.2	93.4\pm0.6	95.6\pm0.5	87.4\pm0.3	92.9\pm0.1

Results The top-3 test accuracy scores (in percentage) are highlighted in Table 2. 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 DATA SETS

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

Data sets 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.

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 \pm 0.70	93.87 \pm 3.33	92.26 \pm 0.71	38.58 \pm 0.25	46.72 \pm 0.46	31.28 \pm 0.27
GAT	76.00 \pm 1.01	71.01 \pm 4.66	78.87 \pm 0.86	35.98 \pm 0.23	63.90 \pm 0.46	42.72 \pm 0.33
APPNP	91.80 \pm 0.63	92.00 \pm 3.59	91.18 \pm 0.70	38.86 \pm 0.24	51.91 \pm 0.56	34.77 \pm 0.34
H2GCN	86.23 \pm 4.71	87.50 \pm 1.77	85.90 \pm 3.53	38.85 \pm 1.77	52.30 \pm 0.48	30.39 \pm 1.22
GCN	66.56 \pm 13.82	66.72 \pm 1.37	75.66 \pm 0.96	30.59 \pm 0.23	60.96 \pm 0.78	45.66 \pm 0.39
Mixhp	60.33 \pm 28.53	77.25 \pm 7.80	76.39 \pm 7.66	33.13 \pm 2.40	36.28 \pm 10.22	24.55 \pm 2.60
GraphSAGE	71.41 \pm 1.24	64.85 \pm 5.14	79.03 \pm 1.20	36.37 \pm 0.21	62.15 \pm 0.42	41.26 \pm 0.26
HRGCN	78.25\pm0.25	91.01\pm1.55	82.25\pm0.91	37.21\pm0.29	56.81 \pm 0.12	44.28 \pm 0.91

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.

7 FINAL REMARK AND CONCLUSION

This paper compared the expressive power between the original GCN and attention based models in terms of their number of linear regions. 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, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning graph embeddings through attention. *CoRR*, abs/1710.09599, 2017. URL <http://arxiv.org/abs/1710.09599>.
- 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*, pp. 21–29. PMLR, 2019.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL <https://arxiv.org/abs/1409.0473>.
- Muhammet Balcilar, Renton Guillaume, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems*, 25(8):1553–1565, 2014.
- Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25):8732–8733, 2009.
- Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.
- Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönaauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.
- 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.
- Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.
- Bhaskar Das Gupta, Marek Karpinski, Nasim Mobasheri, and Farzane Yahyanejad. Effect of gromov-hyperbolicity parameter on cuts and expansions in graphs and some algorithmic implications. *Algorithmica*, 80(2):772–800, 2018.
- Bhaskar DasGupta, Mano Vikash Janardhanan, and Farzane Yahyanejad. Why did the shape of your network change?(on detecting network anomalies via non-local curvatures). *Algorithmica*, 82(7):1741–1783, 2020.
- Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.
- Robin Forman. Bochner’s method for cell complexes and combinatorial ricci curvature. *Discrete and Computational Geometry*, 29(3):323–374, 2003.
- Hervé Fournier, Anas Ismail, and Antoine Vigneron. Computing the gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015.
- Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pp. 35–41, 1977.

- Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pp. 2083–2092. PMLR, 2019.
- Richard S Hamilton. Three-manifolds with positive ricci curvature. *Journal of Differential geometry*, 17(2):255–306, 1982.
- Will Hamilton, Zhithao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pp. 2596–2604. PMLR, 2019.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- Katsuhiko Ishiguro, Shin-ichi Maeda, and Masanori Koyama. Graph warp module: an auxiliary module for boosting the power of graph neural networks. *arXiv preprint arXiv:1902.01020*, 2019.
- Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1):312–320, 2005.
- 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*, pp. 1378–1387. PMLR, 2016.
- John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. pp. 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 Eunye 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*, pp. 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.
- Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 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.
- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.
- 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 (eds.), *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*, pp. 5115–5124, 2017.
- Guido Montúfar, Yue Ren, and Leon Zhang. Sharp bounds for the number of regions of maxout networks and vertices of minkowski sums. *arXiv preprint arXiv:2104.08135*, 2021.
- 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, pp. 4602–4609, 2019.
- Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.
- 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)*, pp. 2758–2766. IEEE, 2015.
- Chien-Chun Ni, Yu-Yao Lin, Jie Gao, and Xianfeng Gu. Network alignment by discrete ollivier-ricci flow. In *International Symposium on Graph Drawing and Network Visualization*, pp. 447–462. Springer, 2018.
- 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 Mathématique*, 345(11):643–646, 2007.
- 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.
- Seongok Ryu, Jaechang Lim, and Woo Youn Kim. Deeply learning molecular structure-property relationships using graph attention neural network. *CoRR*, abs/1805.10988, 2018. URL <http://arxiv.org/abs/1805.10988>.
- 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.
- Jo Schlemper, Ozan Oktay, Michiel Schaap, Mattias Heinrich, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention gated networks: Learning to leverage salient regions in medical images. *Medical image analysis*, 53:197–207, 2019.
- Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pp. 4558–4566. PMLR, 2018.
- Chao Shang, Qingqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. Edge attention-based multi-relational graph convolutional networks. 2018. doi: 10.48550/ARXIV.1802.04944. URL <https://arxiv.org/abs/1802.04944>.
- Dai Shi, Junbin Gao, Xia Hong, ST Boris Choy, and Zhiyong Wang. Coupling matrix manifolds assisted optimization for optimal transport problems. *Machine Learning*, 110(3):533–558, 2021.
- Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

- 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.
- Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.
- Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10296–10305, 2019a.
- Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019b.
- Melanie Weber, Jürgen Jost, and Emil Saucan. Forman-ricci flow for change detection in large dynamic data sets. *Axioms*, 5(4):26, 2016.
- 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*, pp. 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*, pp. 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*, pp. 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.
- Xuebin Zheng, Bingxin Zhou, Junbin Gao, Yu Guang Wang, Pietro Lió, Ming Li, and Guido Montúfar. How framelets enhance graph neural networks. *arXiv preprint arXiv:2102.06986*, 2021.
- Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

A APPENDIX

In the Appendix, we firstly provide the empirical summaries on the rank degeneracy phenomenon via GCN, GAT and HRGCN. Then we show the formal proofs of the previous statements and then show the details of the three experiments mentioned in the paper.

A.1 RANK DEGENERACY PHENOMENON

In this section, we first provide evidence for the rank degeneracy phenomenon for citation networks. Table 4 summarizes the rank of (re-weighted) adjacency matrices from GCN, GAT and HRGCN

Table 4: The rank of (re-weight) adjacency matrices of GCN, GAT and HRGCN

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
Rank of $\eta \odot \hat{A}$	2708	3326	19699	13752	18330	34388	7641

From table 4 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 ($\theta \odot \hat{A}$) and HRGCN ($\eta \odot \hat{A}$) are with much larger number of ranks. Furthermore, the rank of adjacency matrix ($\eta \odot \hat{A}$) in HRGCN has little difference to the number of nodes (which is the maximum possible rank of adjacency matrix) of the dataset, this indicates that our HRGCN not only can utilize the adjacency information from the matrix with repeat rows deleted, but also capable of distinguishing the nodes with the same connectivities. This shows the effectiveness of applying refined Ricci curvature in our model.

A.2 FORMAL PROOFS

Here we show the proof of Lemma 2, That is, in the real practice, the probability of randomly simulate a rank degenerated attention matrix within the space that contains all possible attention matrix is 0.

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 $\mu(S_2) = 0$.*

Proof. It is easy to verify that S_1 defines a manifold \mathcal{M}_1 (multinomial symmetric and stochastic manifold), since all matrices contained in S_1 are symmetric and the summation of each row equals to 1, thus there are maximally $\frac{n(n-1)}{2}$ free elements in the matrices in S_1 . Hence we have the intrinsic dimension of \mathcal{M}_1 equal to $\frac{n(n-1)}{2}$. Similarly, S_2 defines a submanifold $\mathcal{M}_2 \subseteq \mathcal{M}_1$ with its dimension less than \mathcal{M}_1 , this is because with one extra requirement ($\det(M) = 0$) introduced to all matrices in S_2 , the degree of freedom of the matrices in S_1 will be at least decreased by 1. Let μ be a measure defined on S_1 , due to the dimensionality difference, we have all matrices that belong to S_2 as measure 0 and that completes the proof. \square

Based on the claim on Lemma 2, we now verify our statement in Theorem 1. To show the HRGCN can balance the advantage within graph attention models in terms of expressive power.

Theorem 1. *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:*

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

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

Proof. Based on Lemma 2, let S_2 be the set that contains all possible matrices of $\theta \odot \hat{A}$, and we have S_2 is of full rank. Now, define $S_3 := \{M \in \mathbb{R}^{n \times n} | m_{i,j} \geq 0, m_{i,j} = m_{j,i}, \text{Diag}(M) = 1 \forall i\}$ be the set that contain all possible matrices of $\eta \odot \hat{A}$. The diagonal entries of the matrices contained in S_3 are fixed as 1 since based on the definition of the refined Ricci curvature defined in equation (2) when $x_i = x_j$, we have their distance $d(x_i, x_j) = 0$ and this yields $m_{i,i} = \text{Exp}(0) = 1$. Furthermore, compared to the matrices in S_1 defined in lemma 2, matrices in S_3 do not have the row summation property ($\sum_j m_{i,j} = 1 \forall i$). Therefore, each row of the matrices in S_3 still only lost one degree of freedom due to the fixed diagonal values. Based on Lemma 2, one can define another measure μ_2 on S_1 , then we have $\mu(S_3) = 0$. Hence matrices contained in S_3 are of full rank. Then based on Lemma 1 we have $\mathcal{R}_{HRGCN} = \mathcal{R}_{ATT}$, and that completes the proof. \square

We now prove the lemma 3 included in the paper. Since lemma 3 aims to show the propose HRGCN is potentially capable of handling the bottleneck problem mentioned in (Topping et al., 2021) by preventing the long range dependency (negative curvatures) from being diluted in the original GCN. To prove lemma 3, we need the following proposition from (Topping et al., 2021):

Proposition 1 (Theorem 2 in (Topping et al., 2021)). *Given an unweighted graph \mathcal{G} , for any edge $i \sim j$ we have $\kappa(i, j) \geq \text{Ric}(i, j)$.*

Here $\kappa(i, j)$ is the Ollivier Ricci curvature and $\text{Ric}(i, j)$ is the balanced Forman curvature defined in (Topping et al., 2021). Please refer to (Topping et al., 2021) for the details of the proof of this proposition. We note that since we have $\kappa(i, j) \geq \text{Ric}(i, j)$ and the negative balanced Forman curvature has been proved to be responsible for the over-squashing problem, hence when we have $\kappa_{i,j} < 0$ we must have $\text{Ric} < 0$ and this illustrates that the negative $\kappa_{i,j}$ is also responsible to the over-squashing problem. With this conclusion in the mind, we now provide the proof for Lemma 3.

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 $\left\| \frac{\partial h_v^{(\ell+1)}}{\partial x_u} \right\|_2 \leq (\alpha\beta)^{\ell+1} (\mathcal{A}^{\ell+1})_{uv}$.*

Proof. First we see $h_v^{(\ell+1)} = \sigma_\ell(W_\ell^T (H^{(\ell)})^T a_v) = \sigma_\ell(\sum_{i=1}^n a_{vi} W_\ell^T h_i^{(\ell)})$, where we let a_i^\top be the i -th row of matrix \mathcal{A} and a_{ij} be the i, j -th entry of \mathcal{A} . Then by chain rule, we obtain

$$\begin{aligned} \left\| \frac{\partial h_v^{(\ell+1)}}{\partial x_u} \right\|_2 &= \left\| \text{diag}\left(\sigma'_\ell(W_\ell^T (H^{(\ell)})^T a_v)\right) \odot \left(\sum_{i_\ell=1}^n a_{vi_\ell} W_\ell^T \frac{\partial h_{i_\ell}^{(\ell)}}{\partial x_u}\right) \right\|_2 \\ &\leq \alpha \left\| \sum_{i_\ell=1}^n a_{vi_\ell} W_\ell^T \frac{\partial h_{i_\ell}^{(\ell)}}{\partial x_u} \right\|_2 \\ &\leq \alpha^{(\ell+1)} \left\| \sum_{i_\ell, i_{\ell-1}, \dots, i_0} a_{vi_\ell} a_{i_\ell i_{\ell-1}} \cdots a_{i_1 i_0} W_\ell^T W_{\ell-1}^T \cdots W_0^T \frac{\partial h_{i_0}^{(0)}}{\partial x_u} \right\|_2 \\ &= \alpha^{(\ell+1)} \left(\sum_{i_\ell, i_{\ell-1}, \dots, i_1} a_{vi_\ell} a_{i_\ell i_{\ell-1}} \cdots a_{i_1 u} \right) \|W_\ell^T W_{\ell-1}^T \cdots W_0^T\|_2 \\ &\leq (\alpha\beta)^{(\ell+1)} (\mathcal{A}^{\ell+1})_{uv} \end{aligned}$$

where we apply the second inequality recursively to obtain the third inequality. \square

Lemma 3 shows when the derivative of activation functions and the weights are bounded, the sensitivity of the features on the input depend critically on the matrix \mathcal{A} , and this lead us to present the numerical verification (i.e., Table 1) in the main page.

A.2.1 RANDOM PERTURBATION AND OVER-SMOOTHING

In this section, we show the reduction from tiny values of $\epsilon \sim U(0, 0.01)/1000$ s.t. $\epsilon < \min(\eta_{i,j} \odot \hat{a}_{i,j})$ to the non-zero entries of $\eta \odot \hat{A}$ can help HRGCN to enjoy a higher distinguishability (expressive power) than GCN and attention based models, in the meanwhile, let HRGCN have a lower risk of over-smoothing than GCN. We show the advantages from ϵ in the next proposition.

Proposition 2. *Let $\mathcal{R}_{\mathcal{F},\theta}^\epsilon(\text{HRGCN})$ and $\mathcal{R}_{\mathcal{F},\theta}(\text{ATT})$ be the number of linear regions induced from HRGCN and attention based model, respectively. For any fixed input and output feature dimension as d_0 and d_1 , we have:*

$$\mathcal{R}_{\mathcal{F},\theta}^\epsilon(\text{HRGCN}) > \mathcal{R}_{\mathcal{F},\theta}(\text{ATT})$$

Proof. Based on the proof of theorem 1 we have $\mathcal{R}_{\mathcal{F},\theta}(\text{HRGCN}) = \mathcal{R}_{\mathcal{F},\theta}(\text{ATT})$, and the only situation that causes both HRGCN and attention based model lost their distinguishability is a graph or a subset of a graph that is complete and all nodes are with the same features. The tiny perturbation from ϵ to the non-zero entries of $\eta \odot \hat{A}$ addresses this issue by introducing the differences into the re-weighted matrix while preserving the connectivity of such complete graph (subset), thus we have $\mathcal{R}_{\mathcal{F},\theta}^\epsilon(\text{HRGCN}) > \mathcal{R}_{\mathcal{F},\theta}(\text{ATT})$. \square

Now we show that the introduction of ϵ can potentially prevent HRGCN from the over-smoothing problem in the original GCN and GAT model (Cai & Wang, 2020). To show this, we firstly quantify the over-smoothing issue by defining graph Dirichlet energy as follows:

Definition 3 (Graph Dirichlet Energy). *Given node embedding matrix $X^{(l)} = \{x_1^{(l)}, x_2^{(l)} \dots x_N^{(l)}\}^T \in \mathbb{R}^{n \times d_l}$ learned from GCN at l -th layer, the Dirichlet energy $E(X^{(l)})$ is defined as:*

$$E(X^{(l)}) = \text{Tr}(X^{(l)T} \tilde{\Delta} X^{(l)}) = \frac{1}{2} \sum_{i,j} w_{i,j} \left\| \frac{x_i^{(l)}}{\sqrt{1+d_i}} - \frac{x_j^{(l)}}{\sqrt{1+d_j}} \right\|_2^2$$

Where $\tilde{\Delta} = I - \hat{A}$ is the normalized graph Laplacian and \hat{A} is the normalized adjacency matrix. The graph Dirichlet energy shows how smooth the information propagate in terms of GNN computation, and it has been reckoned as one of the metric that measures the over-smoothing issue in both GCN and GAT (Cai & Wang, 2020). Specifically, recall the computation within GCN can be described as:

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

If one were to remove the activation function σ , we have $\lim_{l \rightarrow \infty} \hat{A}^l H^{(0)} = H^{(\infty)}$, where each row of $H^{(\infty)}$ only depends on the degree of the corresponding node, meaning that the graph node features produced from the prior layer is irreducible and aperiodic. Thus the learning model loses discriminative information provided by the node features as the number of layers increases. Thanks to the next theorem we can show that by reducing ϵ to the product of $\eta \odot \hat{A}$, HRGCN produces a higher Dirichlet energy than GCN within any finite layers.

Theorem 2. *Let $\tilde{A} = \eta \odot \hat{A}$ and $\tilde{A}_\epsilon = \eta \odot \hat{A} - \epsilon$ be the re-weighted matrices of the curvature matrix η and the perturbed curvature matrix η_ϵ , respectively. Let $\tilde{\Delta}$ and $\tilde{\Delta}_\epsilon$ be the Laplacian matrices induced from \tilde{A} and \tilde{A}_ϵ , respectively. Then for any $\epsilon > 0$ and $\epsilon < \min(\eta_{i,j} \odot \hat{a}_{i,j})$, at any specific layer (i.e., l -th layer), we have:*

$$E_\eta(X^{(l)}) < E_{\eta_\epsilon}(X^{(l)})$$

Where $E_\eta(X^{(l)})$ and $E_{\eta_\epsilon}(X^{(l)})$ are the Dirichlet energy at layer k induced from η with and without perturbation ϵ .

Proof. The result can be easily proved by verification since we have:

$$\tilde{A} = \eta \odot \hat{A} \quad \text{and} \quad \tilde{A}_\epsilon = \eta \odot \hat{A} - \epsilon$$

Then we have:

$$\tilde{\Delta} = I - \eta \odot \hat{A} \quad \text{and} \quad \tilde{\Delta}_\epsilon = I - \eta \odot \hat{A} + \epsilon$$

For the perturbed graph Laplacian $\tilde{\Delta}_\epsilon$ we have:

$$\begin{aligned} E(X^{(l)})_\epsilon &= \text{Tr}(X^{(l)T} \tilde{\Delta}_\epsilon X^{(k)}) = \text{Tr}(X^{(l)T} (I - \eta \odot \hat{A} + \epsilon) X^{(l)}) \\ &= \text{Tr}(X^{(l)T} (\tilde{\Delta} + \epsilon) X^{(l)}) = \text{Tr}(X^{(l)T} \tilde{\Delta} X^{(l)}) + \text{Tr}(X^{(l)T} \epsilon X^{(l)}) \\ &= \text{Tr}(X^{(l)T} \epsilon X^{(k)}) + E(X^{(l)}) \end{aligned}$$

Since $\epsilon > 0$ thus we have $\text{Tr}(X^{(l)T} \epsilon X^{(l)}) > 0$ and therefore we have an positive increase of Dirichlet energy from ϵ \square

Hence we have proved that with the help of the random perturbation that initially assigned to HRGCN to increase its expressive power, we also lift system’s Dirichlet energy to make HRGCN robust to over-smoothing.

A.3 EXPERIMENT EXTEND

The code for this paper can be found at <https://github.com/dshi3553usyd/HRGCN-high-rank-GCN-.git>.

A.3.1 CURVATURE ASSISTED GRAPH POOLING

In this section, we show numerical results on (refined) curvature assisted graph pooling. Specifically, recall the self-attention graph pooling model (Lee et al., 2019) in which the attention score is generated as: $Z = \sigma(\hat{A}X\theta_{att})$, Where \hat{A} is the normalized adjacency matrix and $\theta_{att} \in \mathbb{R}^{d_0 \times 1}$ is the attention coefficient matrix learned by the model. Since we have shown that HRGCN can produce the identical expressive power compared to attention based models, thus the refined Ricci curvature can naturally enhance the graph pooling schemes by illustrating the graph topological importance in terms of information (pooling) aggregation. Therefore, the curvature-based graph pooling model can be formulated as: $Z = \sigma(\eta \odot \hat{A}X)$, where $\eta_{i,j} = \text{Exp}(-\tilde{\kappa}_{i,j})$, similar to attention pooling model, the pooling ratio $k \in (0, 1]$ is a hyperparameter that determines the number of nodes to keep. The top $[kn]$ nodes are selected based on the value of Z . Finally we equip the curvature pooling strategy into the HRGCN model and therefore the final attention score for HR_Pool can be expressed as: $Z = \sigma(\text{HRGCN}(X, \hat{A}))$.

Dataset Six benchmarks were selected to test the prediction power of HR_Pool, including four classification tasks with moderate sample size, one large scale classification task and one regression task. The classification tasks use the **TUDataset benchmarks** (Morris et al., 2020) including **D&D** (Dobson & Doig, 2003), **PROTEINS** (Borgwardt et al., 2005) to categorize proteins into enzyme and non-enzyme structures; **NC11** (Wale et al., 2008) to identify chemical compounds that block lung cancer cells; **Mutagenicity** (Kazius et al., 2005) to recognize mutagenic molecular compounds for potentially marketable drug; and **QM7** (Blum & Reymond, 2009) to predict atomization energy value of molecules. The rest, namely **ogbn-molhiv** (Hu et al., 2020) is used for large scale molecule classification.

Setup All the baseline models are with two fixed convolutional layers followed by one pooling layer as the network architecture. The graph convolution for the five **TUDatasets** uses the GCN model, and for **ogbg-molhiv** uses GIN with virtual nodes (Ishiguro et al., 2019). Given graph representations, the prediction is made by a two-layer MLP, in which the hidden unit is identical to that of the convolutional layer. The parameters, including learning rate, weight decay, number of hidden unit in the convolutional layer and drop out ratio, are fine-tuned using grid search mentioned earlier. The dataset was also split using standard data splitting method as the benchmark models did. Similar to the method mentioned in (Zheng et al., 2021), the training stops when the validation loss stops improving for 20 consecutive epochs or reaching maximum 200 epochs. The accuracy results are averaged over 10 repetitions. For **TUDataset**, the mean test accuracy is reported, and for **ogbg-molhiv**, ROC-AUC score is used. The regression task on QM7 is reported as mean square error (MSE).

Baseline The learning outcome of RC-Pooling models are compared to seven baseline methods. These baselines are TOPKPool (Gao & Ji, 2019), ATTENTIONPool (Li et al., 2020), SAGPool (Lee et al., 2019), (Zheng et al., 2021), and the classic SUM, MAX and MEAN pooling.

Table 5: Performance comparison between graph property prediction models. **QM7** is a regression task evaluated by MSE; **ogbn-molhiv** task by AUC-ROC percentage; others datasets are for classification and evaluated by test percentage accuracy. The values after \pm are standard deviations. The top results are highlighted in **bold**.

Datasets	PROTEINS	Mutagenicity	D&D	NCI1	ogb-molhiv	QM7
TOLPool	73.48 \pm 3.57	79.84 \pm 2.46	74.87 \pm 4.12	75.11 \pm 3.45	78.14 \pm 0.62	175.41 \pm 3.16
ATTENTION	73.93 \pm 5.37	80.25 \pm 2.22	77.48 \pm 2.65	74.04 \pm 1.27	74.44 \pm 2.12	177.99 \pm 2.22
SAGPool	75.89 \pm 2.91	79.86 \pm 2.36	74.96 \pm 3.60	76.30 \pm 1.53	75.26 \pm 2.29	41.93 \pm 1.14
SUM	74.91 \pm 4.08	80.69 \pm 3.26	78.91 \pm 3.37	76.96 \pm 1.70	77.41 \pm 1.16	42.09 \pm 0.91
MAX	73.57 \pm 3.94	78.83 \pm 1.70	75.80 \pm 4.11	75.96 \pm 1.82	78.16 \pm 1.33	177.48 \pm 4.70
MEAN	73.13 \pm 3.18	80.37 \pm 2.44	76.89 \pm 2.23	73.70 \pm 2.55	78.21 \pm 0.90	177.49 \pm 4.69
HR_Pool	76.77\pm2.15	81.49\pm3.15	77.50 \pm 2.21	77.1\pm3.25	79.40\pm2.51	150.24\pm3.25

Results From Table 3 we can see the the propose pooling model in this paper outperforms the attention based pooling model in terms of both graph-level regression and classification tasks.

A.3.2 SUMMARY STATISTICS OF THE DATASETS

In this section, we show some statistics of the graph datasets mentioned in the paper, and provide the sensitivity analysis on the hyperparameter α which is the initial mass assigned onto each node of the graph. As the graph Ricci curvature illustrates the connectivity importance between nodes, when $\alpha \approx 1$, indicating most of the initial mass are assigned to the nodes itself, causing the Wasserstein distance approaching to the shortest distance if the graph is unweighted and thus $\kappa_{i,j} = 1 - \frac{W_{i,j}}{d_{i,j}} \approx 0$. On the other hand, when $\alpha \approx 0$, the connectivity importance based on the $W_{i,j}$ value gradually appears. In the next few tables we show the benchmark statistics on (homophily) citation networks, (heterophily) benchmarks and datasets for graph Pooling. Moreover, similar to (Ye et al., 2019), the computational complexity and time of the refined Ricci curvature for citation networks are also included. In addition, we also provide the hyperparameter searching spaces for both node classification and graph pooling.

Hyperparameter Tuning Space We tuned the hyperparameters with the following selection of values. For learning rate: {0.1, 0.05, 0.01, 0.005}, number of hidden units in {16, 32, 64, 96}, weight decay in {0.001, 0.005, 0.01, 0.05} and scale in {0.1, 0.5, 0.7, 0.9} for **Cora**, **Citeseer** and **Pubmed**, {7, 8, 9, 10} for **CS**, **Physics**, **Computers** and **Photo**. For the homophily graph and the graph dataset used in pooling we fixed Ollivier $\alpha = 0.9$ whereas for heterophily graphs we fixed $\alpha = 0.4$. The following tables shows the summary statistics of the datasets experimented in the paper.

Table 6: Summary statistics for homophily citation networks. Moreover, the computational time for curvatures in these networks are: 1.99s, 2.38s, 20.8s, 39.5s, 312.8s, 620s and 820s.

Datasets	#Classes	#nodes	#Edges	#Features	#Training	#Edges/#Nodes
Cora	7	2708	5429	1433	140	2.0
Citeseer	6	3327	4372	3703	120	1.42
Pubmed	3	19717	44338	500	60	2.25
Coauthor CS	15	18333	100227	6805	300	5.47
Coauthor Physics	5	34493	495924	8415	100	14.37
Amazon Computer	10	13381	259159	767	200	19.37
Amazon Photo	8	7487	126530	745	150	16.90

Table 7: Summary Statistics of the datasets, $H(G)$ represent the level of homophily of overall benchmark datasets

Datasets	#Class	#Feature	#Node	#Edge	Training	Validation	Testing	H(G)
Chameleon	5	2325	2277	31371	60%	20%	20%	0.247
Squirrel	5	2089	5201	198353	60%	20%	20%	0.216
Film	5	932	7600	26659	60%	20%	20%	0.221
Wisconsin	5	251	499	1703	60%	20%	20%	0.150
Texas	5	1703	183	279	60%	20%	20%	0.097
Cornell	5	1703	183	277	60%	20%	20%	0.386

Datasets	PROTEINS	Mutagenicity	D&D	NCI1	ogbg-molhiv	QM7
#Graphs	1113	4337	1178	4110	41127	7165
Min #Nodes	4	4	30	3	2	4
Max # Nodes	620	417	5748	111	222	23
Avg#Nodes	39	30	284	30	26	15
Avg#Edges	73	31	716	32	28	123
#Features	3	14	89	37	9	0
#Classes	2	2	2	2	2	1(R)

Table 8: Summary statistics for the Graph Pooling Benchmarks, the letter R in the class number of QM7 represents a regression task

A.3.3 COMPUTATIONAL COMPLEXITY FOR GRAPH RICCI CURVATURE

The exact computation of graph Ricci curvature for large graph is somehow time costly since a learning programming problem need to be solved on each edge of the graph. Based on (Ye et al., 2019), on each edge, to obtain the Wasserstein distance between the distributions generated from the probability measure function, the learn programming is conducted with $d_x \times d_y$ variables and $d_x + d_y$ constraints. Using the interior point solver (ECOS), the complexity is $\mathcal{O}((d_x \times d_y)^w)$ in which w is the exponent of the complexity of matrix multiplication (the best known is 2.373). However, there are many approximation methods that can relax the computation of optimal transportation such as Sinkhorn Algorithm (Cuturi, 2013) and some methods can increase the precision of the Wasserstein distance for example (Shi et al., 2021) and has proved to have almost identical computational complexity to the classic OT algorithms. We included the computation time for citation networks in Table 4.