Curvature Constrained MPNNs: Improving Message Passing with Local Structural Properties

Hugo Attali, Davide Buscaldi, Nathalie Pernelle

Curvature Constrained MPNNs : Improving Message Passing with Local Structural Properties

Hugo Attali, Davide Buscaldi, Nathalie Pernelle

[a]*LIPN, UMR 7030, CNRS, University Sorbonne Paris Nord, Villetaneuse, 93430, France*

Abstract

Graph neural networks operate through an iterative process that involves updating node representations by aggregating information from neighboring nodes, a concept commonly referred to as the message passing paradigm. Despite their widespread usage, a recognized issue with these networks is the tendency to over-squash, leading to diminished efficiency. Recent studies have highlighted that this bottleneck phenomenon is often associated with specific regions within graphs, that can be identified through a measure of edge curvature. In this paper, we present a novel framework designed for any Message Passing Neural Network (MPNN) architecture, wherein information distribution is guided by the curvature of the graph's edges. Our approach aims to address the over-squashing problem by strategically considering the geometric properties of the underlying graph. The experiments carried out show that our method demonstrates significant improvements in mitigating over-squashing, surpassing the performance of existing graph rewiring techniques across multiple node classification datasets.

Keywords: Graph Neural Netwoks, over-squashing, rewiring, discrete curvature

1. Introduction

Graph representation learning is a rapidly expanding research field that focuses on the development of versatile methods for effectively learning representations from graph-structured data (Goller and Kuchler, 1996) (Gori et al., 2005) (Scarselli et al., 2008) (Bruna et al., 2014). The majority of Graph neural networks *GNNs* are based on the message passing paradigm (Gilmer et al., 2017), in which the information is propagated by the iterative

exchange of information (messages) between neighboring nodes to update their representations. This process is typically performed over multiple iterations and/or layers. The message passing paradigm is effective in capturing the relational information and structural patterns within graph-structured data. It enables GNNs to learn expressive representations that are sensitive to the connectivity and interactions among nodes in a graph. GNNs have been successful in various domains, including chemistry, information retrieval, social network analysis, and knowledge graphs, due to the wide variety of features that a graph can model (Wu et al., 2020). These architectures have produced very interesting results when it comes to solving tasks at the node, graph, or edge level (Xiao et al., 2022) (Errica et al., 2020) (Zhang and Chen, 2018).

Despite their widespread use, it has been shown that GNNs can face a variety of issues under certain conditions, specifically in heterophilic environments (Zhu et al., 2020) (Platonov et al., 2023), when the neighboring nodes tend to have different labels. Other works have highlighted that GNNs suffer from a limited ability to model long-range interactions (Alon and Yahav, 2021). Popular GNN architectures such as Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) and Graph Attention Networks (GAT) (Veličković et al., 2018) can only share information between nodes at a distance that depends on the number of layers in the architecture: for a node i to be able to exchange information with a node $j \in \mathcal{N}_k(i)$, we need to stack at least k layers. Therefore, a naive approach to address this issue consists of increasing the number of layers.

However, this process leads to two well-known problems for GNN. First, the phenomenon of over-smoothing which arrives when the message passing is carried out in an excessive way. In this case, all the features of the nodes are going to be similar wich leads to a deterioration in results (Oono and Suzuki, 2020) (Cai and Wang, 2020). Second, as the number of layers in a GNN grows, information from exponentially growing receptive fields must be propagated concurrently at each message-passing step, leading to a bottleneck that causes over-squashing (Alon and Yahav, 2021). In this case, spreading information locally is not enough. To overcome this problem, GNNs must be able to incorporate additional global graph features in the process of learning representations. Another popular approach is to rewire the input graph to improve the connectivity and avoid over-squashing.

Recently, it has been shown that the local structural properties of a graph, like edge curvature, play an essential component in the spread of knowledge about the graph (Topping et al., 2022).

Main Contributions. This paper introduces an innovative framework applicable to any Message Passing Neural Network (MPNN) architecture, able to mitigate over-squashing. The main contributions are:

- Introducing a novel metric for homophily, based on edge curvature, that allows to better model the neighborhood community behavior.

- Motivated by this metric we propose a new MPNN architecture (Curvature-Constrained Message Passing) that leverages the curvature of the edges to guide learning by dissociating edges with positive and negative curvature. We propose different variants of this model, each one based on a different way of propagating the information: only on edges with negative curvature, or positive curvature. We also propose two- or one-hop propagation strategies that are bound to the curvature.

- We empirically demonstrate a performance gain on heterophilic datasets and we show that using a curvature message passing attenuates over-squashing.

Reproducibility.. Our codes to reproduce the experiments of the paper is available. [1]

2. Related Work

In this section, we introduce the class of Message Passing Neural Networks and discuss some of its main limitations. We first review some important notions concerning graphs.

[1] Code available from: `https://github.com/Hugo-Attali/Curvature-Constrained-Message-Passing`

2.1. Preliminaries

We start by introducing notations used throughout this paper. A graph is written as a tuple $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges, respectively. In this work, we focus on undirected graphs, i.e., if $e_{ij} \in \mathcal{E}$, then $e_{ji} \in \mathcal{E}$. We note by $\mathcal{N}(i)$ the set of neighbors of node i. We define $N \times N$ adjacency matrix $\mathbf{A}$ as $\mathbf{A}_{i,j} = 1$ if $(i, j) \in \mathcal{E}$ and zero otherwise. We add self-loops to each node such that $\hat{A} = \mathbf{A} + \mathbf{I}$. In addition, we let $\mathbf{D}$ be the diagonal matrix and $L = I - D^{-1/2}AD^{-1/2}$ be the normalized Laplacian of G. We denote the nodes feature by H where h_i is the feature of node i.

2.2. Message Passing Neural Networks

The remarkable achievements of deep learning within the Euclidean domain have spurred significant interest in extending the capabilities of neural networks to non-Euclidean domains, such as graphs.

The main objective of the message passing approach is to iteratively find an effective node embedding that captures context and neighborhood information (Gilmer et al., 2017). The message passing technique consists of two phases which iteratively apply the AGGREGATE and UPDATE function to compute embeddings h_i^ℓ at the layer ℓ based on message $m_i^{(\ell)}$ containing information on neighbors :

$$\begin{aligned} m_i^{(\ell)} &= \text{AGGREGATE}^{(\ell)}\left(h_i^{(\ell-1)}, \left\{h_j^{(\ell-1)}\ j \in \mathcal{N}(i)\right\}\right), \\ h_i^\ell &= \text{UPDATE}^{(\ell)}\left(h_i^{(\ell-1)}, m_i^{(\ell)}\right) \end{aligned} \tag{1}$$

For GCN (Kipf and Welling, 2017) $m_i^\ell = \sum_{j \in \mathcal{N}(i)} \frac{\mathbf{h}_j^\ell}{\sqrt{|\mathcal{N}(i)||\mathcal{N}(j)|}}$,

while for GAT (Veličković et al., 2018) $m_i^\ell = \sum_{j \in \mathcal{N}(i)} a_{ij}^\ell\, \mathbf{h}_j$, with :

$$a_{ij}^{(\ell)} = \frac{\exp\left(\text{LeakyReLU}(z^{(\ell)} \cdot [h_i^{(\ell-1)}\mathbf{W}^{(\ell)} || h_j^{(\ell-1)}\mathbf{W}^{(\ell)}])\right)}{\sum_{j \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}(z^{(\ell)} \cdot [h_i^{(\ell-1)}\mathbf{W}^{(\ell)} || h_j^{(\ell-1)}\mathbf{W}^{(\ell)}])\right)}$$

where $||$ stands for concatenation. This score is parameterised by $z^{(l)}$ and $\mathbf{W}^{(l)}$, respectively a weight vector and a linear transformation.

As classical MPNNs only send messages along the edges of the graph, this will prove particularly interesting when adjacent nodes in the graph

share the same label (homophilic case). On the other hand, working in a heterophilic environment with classical MPNNs can leads to low performance (Zheng et al., 2022). Indeed one of the main drawbacks of classical MPNNs is to rely only on one-hop message propagation. Additional layers must be stacked to capture non-local interactions. However, this leads to over-squashing discussed in the section 2.5.

2.3. Graph Curvature

As for a manifold, the notion of curvature is a good way to classify the local behavior of a graph. The Ricci curvature of a manifold can be characterized by the concept of "geodesic dispersion," indicating whether two parallel geodesics originating from nearby points converge, remain parallel (Euclidean case), or diverge (indicating negative curvature and giving rise to hyperbolic geometry). To analogize geodesic dispersion on graphs, consider an edge e_{ij} and two edges emanating from nodes i and j. To draw the analogy with discrete spherical geometry, these edges would intersect at another node, forming a triangle (a 3-clique). In a discrete Euclidean geometry, the edges would persist in parallel, creating a rectangle (4-cycle) based on e_{ij} (a grid). Finally, in a discrete hyperbolic geometry, the mutual distance between the ends of the edges would increase concerning that of i and j, related to the structure of a tree.

More generally discrete graph curvature describes how the neighbors of two nodes are structurally connected. (Forman, 2003) and (Ollivier, 2007) were the first to propose a measure of discrete graph curvature. Numerous studies have demonstrated the usefulness of edge curvature for various graph tasks. For instance (Jost and Liu, 2014) (Ni et al., 2019) (Sia et al., 2019) use Ollivier curvature for community detection.
Another work (Ye et al., 2019) defined Curvature Graph Neural architecture which calculates an attention mechanism based on Ollivier curvature. They demonstrate the benefits of such an architecture for the task of node classification. More recently, (Jost and Liu, 2014) and (Topping et al., 2022) proposed extensions to Forman's curvature to improve its expressiveness. (Topping et al., 2022) demonstrate the correlation between edge curvature and over-squashing phenomenon. In this paper, we focus on the best known discrete curbature, Ollivier curvature (Ollivier, 2007), Augmented Forman Curvature and Balanced Forman Curvature as detailed in Section 2.4.

Illustration of graph curvature are provide in Figure 1.

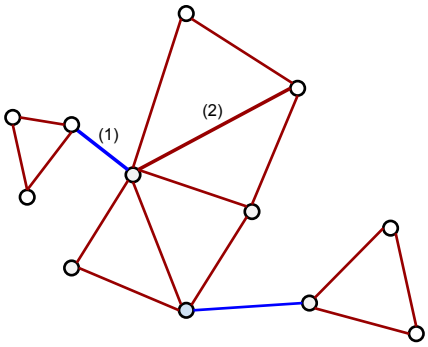

Figure 1: In red, edges with positive curvature,connect nodes in the same community, and in blue, edges with negative curvature connect nodes in different communities. Edge Curvature Measures using Augmented Forman Curvature (AFC) and Ollivier Curvature (O) on edge (1) and (2) is $AFC((1)) = -4$ and $AFC((2)) = 1$ $O((1)) = -0.5$ and $O((2)) = 0.25$.

2.4. Curvature measures

In this paper, we use the curvature of graphs by employing three established definitions of edge curvature in the context of information diffusion: Ollivier curvature (Ollivier, 2007), Augmented Forman Curvature (Samal et al., 2018), Balanded Forman Curvature (Topping et al., 2022). We provide a exemple of edge curvature value on Figure 1.

Ollivier Curvature. Let's define a probability distribution μ_i over the nodes of the graph such that we apply to each node i a lazy random walk probability measure α :

$$\mu_i : j \mapsto \begin{cases} \alpha & if \quad j = i \\ (1-\alpha)/d_i & if \quad j \in \mathcal{N}(i) \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

Following previous work (Ni et al., 2015) (Ni et al., 2018) we choose $\alpha = 0.5$. We then consider the Wasserstein distance of order 1, $W_1(i,j)$, corresponding to the optimal transport of probability masses from i neighbors to j neighbors.

$$W_1(\mu_i, \mu_j) = \inf_{\alpha \in \Pi(\mu_i, \mu_j)} \sum_{i,j \in V} dist(i,j)\, M(i,j) \tag{3}$$

where $\Pi(\mu_1, \mu_2)$ denotes the set of probability measures with marginals μ_i and μ_j. where $M(i, j)$ is the amount of mass moved from i to j along the shortest path of i and j.
Finally, the Ollivier curvature c_{ij} of an edge e_{ij} can be defined as :

$$c_{ij} = 1 - \frac{W_1(\mu_i, \mu_j)}{dist(i, j)}, \tag{4}$$

where $dist(i, j)$ is the shortest path beetween node i and node j.

Augmented Forman Curvature. The curvature measure proposed by (Samal et al., 2018) proposes to extend Forman's curvature taking into account the triangles in the graph.

For an undirected graph :

$$c_{ij} = 4 - D_{ii} - D_{jj} + 3m, \tag{5}$$

where m is the number of triangles that contain e_{ij}.

Balanced Forman Curvature. (Topping et al., 2022) proposes a more expressive combinatorial measure than augmented Forman, which can only distinguish triangles considering cylce :

$$\begin{aligned} c_{ij} = \frac{2}{D_{ii}} + \frac{2}{D_{jj}} - 2 + 2\frac{m}{\max\{D_{ii}, D_{jj}\}} + \\ \frac{m}{\min\{D_{ii}, D_{jj}\}} + \frac{(\Gamma_{\max})^{-1}}{\max\{D_{ii}, D_{jj}\}}(\gamma_i + \gamma_j) \end{aligned} \tag{6}$$

where $\Gamma_{\max}(i, j)$ is the maximal number of 4-cycles based at e_{ij}, and γ_i is the number of 4-cycles based at e_{ij} without diagonals inside.

2.5. Over-squashing

Long-range tasks need the propagation of information across several levels. The node representations are aggregated with others at each stage before being passed on to the next node. Because the size of the node feature vectors remains constant, they rapidly exhaust their representational capacity in order to retain all of the previously integrated information. When an exponentially expanding quantity of information is squashed into a fixed-size

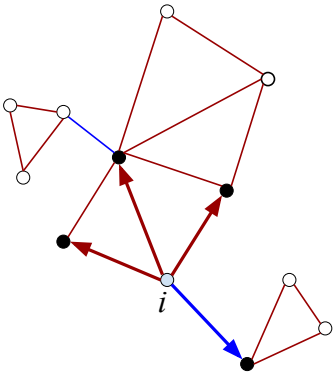

Figure 2: A

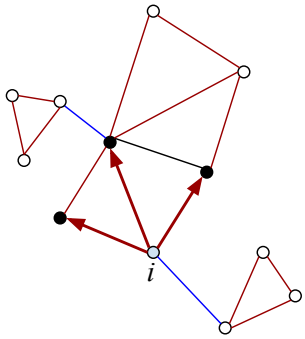

Figure 3: C

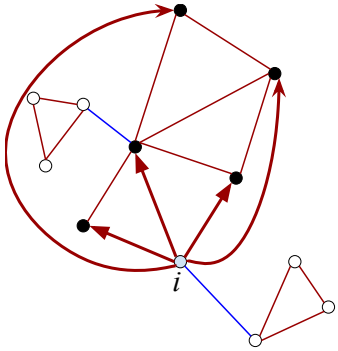

Figure 4: D

Figure 5: On the left a classic message passing for the first (A) starting from a given node i. On the right an example of one-hop positive curvature (C) and two-hop positive curvature (D) message passing. The message is propagated not only to the adjacent nodes but also to those at distance two along positively curved edges i.e following a chain of positively curved edges of size 2.

vector, over-squashing happens (Alon and Yahav, 2021).

To quantify this phenomenon, some approaches exploit the spectral gap (Banerjee et al., 2022) (Karhadkar et al., 2023), which is closely linked to the Cheeger constant (Chung and Graham, 1997).

$$Ch(G) = \min_{1 \leq |S| \leq \frac{|V|}{2}} \frac{|\partial S|}{|S|}, \tag{7}$$

with $S \subset V$ and where $\partial S = \{(i,j) : i \in \mathcal{S}, j \in \overline{\mathcal{S}}, (i,j) \in E\}$

If Cheeger's constant is small, there is a bottleneck structure in the sense that there are two large groups of vertices with few edges connecting them. Cheeger's constant is large if a feasible vertex split into two subsets has "many" edges between these two subsets.
Calculating the precise value of $Ch(G)$ is too costly. The discrete Cheeger inequality (Alon and Milman, 1984) (Cheeger, 1970) shows the link between the spectral gap and the Cheeger constant. The spectral gap of G is the difference between the first two eigenvalues λ_2 - λ_1 of L with $\lambda_1 = 0$.

$$\frac{\lambda_2}{2} \leq Ch(G) \leq \sqrt{2\lambda_2} \tag{8}$$

To mitigate the over-squashing phenomenon different works have proposed various methods to improve the local connectivity of the graph.

Rewiring methods. Most methods address over-squashing by rewiring the input graph i.e. modifying the original adjacency matrix such that it has fewer structural bottlenecks. The work (Alon and Yahav, 2021) were the first to highlight the problem of GNN over-squashing. They propose to modify the GNN's last layer in order to connect all of the nodes. (Topping et al., 2022) shows that the highly negatively curved edges are characteristic of the bottleneck phenomenon and therefore disrupt message passing. They propose a stochastic discrete Ricci Flow (SDRF) rewiring approach, which tries to raise the balanced Forman curvature of negatively curved edges by adding and removing edges. BORF, as presented by (Nguyen et al., 2023), establishes a connection between edges exhibiting very high positive curvature. Consequently, it suggests a strategy of removing and adding edges to avoid edges with strongly negative and positive curvatures, aiming to simultaneously mitigate both over-squashing and over-smoothing. This is achieved by leveraging the curvature of Ollivier. (Karhadkar et al., 2023) propose an algorithm (FOSR) for adding edges at each step to maximize the spectral gap. Because calculating the spectral gap for each edge addition is costly, FOSR employs a first-order spectral gap approximation based on matrix perturbation theory.
Without the direct objective of reducing the phenomenon of over-squashing, other methods such as (Klicpera et al., 2019) modify the adjacency matrix to improve the connectivity of the graph (DIGL). This method adds edges based on the PageRank algorithm, followed by sparsification. As PageRank works using random walks, DIGL tends to improve the connectivity among nodes in the intra-community parts of the graph.

Master node. Another way to reduce over-squashing consists of the introduction of a sort of "global context" by introducing a master node. This node is connected to all other nodes in the graph (Battaglia et al., 2018) (Gilmer et al., 2017). Since the hop distance between all other nodes is at a maximum of two, the reduction of over-squashing is assured (except for the master node). However, in large graphs, incorporating information over a very large neighborhood leads to poor quality of the master node embedding.

Expander Graphs. Another work (Deac et al., 2022) adopt a strategy based on expander graphs, adding to the GNN a layer based on a Cayley graph of the same size as the original input graph. These graphs have some desirable properties, such as being sparse and having a low diameter. The smaller diameter means that any two nodes in the graph can be reached in a reduced number of hops, which removes bottlenecks.

3. Curvature Message Passing

We propose a novel method that introduces a homophily measure constrained by graph edge curvature, enabling selective information flow along edges with distinct curvature values. Our main contribution is the development of a flexible framework that dynamically reshapes the graph by strategically adding or filtering edges to enhance homophily. This approach not only improves information propagation but also facilitates efficient network restructuring. Moreover, our method is compatible with any Message Passing Neural Network (MPNN) architecture, making it a versatile tool for refining node features through the use of curvature information.

Our framework integrates three existing methods for calculating graph edge curvature but can also accommodate alternative structural indicators, offering added flexibility for analysis. In this section, we first introduce a new homophily measure that incorporates edge curvature, followed by a detailed presentation of the Curvature-Constrained Message Passing framework.

3.1. Curvature-Constrained Homophily

The homophily of a graph has a determining role in the efficiency of architectures on a node classification task. Many homophily measures exist in literature (Pei et al., 2020) (Zhu et al., 2020) (Lim et al., 2021) (Platonov et al., 2022); the most commonly used are node homophily (Pei et al., 2020) which computes the average of the proportion of neighbors that have the same class y for each node and edge homophily β (Zhu et al., 2020) which corresponds to the fraction of edges that connect nodes of the same class:

$$\beta = \frac{|\{(i,j) : (i,j) \in E \wedge y_i = y_j\}|}{|E|}$$

The main limitation of this measure is that it doesn't fully capture the local structural characteristics of the graphs. Therefore, we propose a new measure of homophily that takes into account the curvature of edges such that:

$$\beta^+ = \frac{|\{(i,j) : (i,j) \in E^+ \wedge y_i = y_j\}|}{|E^+|}$$

Where E^+ is the set of edges (i,j) such that $c_{ij} \geq \epsilon$. β^- is conversely defined using E^-, the set of edges (i,j) such that $c_{ij} < \epsilon$. A high value for the positive curvature homophily means that the fraction of edges that connect nodes within the same community tend to have the same label.
The values of β^+ and β^- derived from the datasets are presented in Table 1 (refer to Table 2 for specific dataset details) using Ollivier-curvature-constrained homophily. In our experiments, we set $\epsilon = \mu$ for both one-hop and two-hop neighborhoods, where μ represents the mean curvature of the graph's edges. We also provide the max homophilic gain relative to the initial metrics (Zhu et al., 2020). Please note that details on Augmented Forman curvature-constrained homophily and Balanced Forman curvature-constrained homophily can be found in Appendix Appendix A.2.

	Dataset	β	β^+	β^-	2-hopβ^+	2-hopβ^-	Max Homophilic Gain
Heterophilic	Squirrel	0.23	0.24	0.25	0.21	0.23	7%
	Chameleon	0.26	0.30	0.26	0.33	0.29	28%
	Texas	0.31	0.43	0.45	0.52	0.46	69%
	Wisconsin	0.36	0.42	0.53	0.44	0.36	48%
	Cornell	0.34	0.47	0.50	0.33	0.36	46%
	Romain-empire	0.29	0.43	0.48	0.05	0.07	67%
	Actor	0.32	0.37	0.42	0.21	0.21	30%
Homophilic	Cora	0.84	0.94	0.82	0.90	0.72	11%
	Citeseer	0,81	0.86	0.83	0.78	0.76	10%
	Photo	0.84	0.93	0.72	0.93	0.51	11%
	Computers	0.78	0.90	0.64	0.90	0.59	16%

Table 1: Comparison of edge homophily measures. The last column reports the max gain in homophily obtained by using the curvature-constrained edge homophily as opposed to edge homophily.

In homophilic datasets, we observe that $\beta^+ \geq \beta$, indicating that intra-community nodes in the graph tend to exhibit more similar labels (on positive curvature edges) than their inter-community counterparts (on negative curvature edges).
In the case of heterophilic datasets, leveraging positively curved edges does not enhance homophily. As suggested in (Zhu et al., 2020), for heterophilic graphs, the 2-hop scenario is more homophilic than the 1-hop. However, our

analysis reveals that, based on curvature-constrained edge homophily, adopting a two-hop neighborhood is significantly advantageous only for smaller datasets, such as the *WebKB dataset*. Notably, negative curvature homophily generally surpasses the initial metrics for heterophilic datasets.

3.2. Curvature-Constrained Message Passing

Based on the previously introduced curvature-constrained homophily measures, we propose to dissociate the spread of information in the graph based on the curvature of the graph edges. We consider diffusions for both one-hop and two-hop connectivity (see the examples in Figure 5).
We propose to extend the aggregation part of the classic MPNNs 1 :

$$h_i^{(\ell)} = \text{UPDATE}^{(\ell)}\left(h_i^{(\ell-1)}, \text{AGGREGATE}^{(\ell)}\left(\left\{h_j^{(\ell)} : j \in \mathcal{N}^+(i)\right\}\right)\right) \quad (9)$$

Where $\mathcal{N}^+$ represents the neighborhood of nodes that are connected by a positively curved edge to i. Diffusing information only on edges with positive curvature allows information to be exchanged only within the communities of the graph. Based on the curvature-constrained homophily used positive curvature adjacency matrix can be useful on homophilic datasets.
Similarly, $Curv_{m_i}^-$ is defined in the same way by considering $\mathcal{N}^-$ instead of $\mathcal{N}^+$. As discussed by (Deac et al., 2022) working with negatively curved edges may seem counterintuitive in relation to the recommendation to avoid negatively curved edges (Topping et al., 2022). We confirm the results of (Deac et al., 2022) by showing empirically that diffusing information through negatively curved edges improves performance and mitigates the oversquashing phenomenon.
Propagating information through the curvature of edges offers greater flexibility in learning representation. For a two-layer GNN, we can either only exchange information on edges with negative or positive curvature *i.e.* using one curvature adjacency matrix, or first broadcast information on edges with positive and then negative curvature, or using both curvature adjacency matrix for the two different layers.

One-hop curvature. Utilizing a one-hop curvature neighborhood allows us to selectively distribute information solely to edges exhibiting a particular curvature. By excluding either edges with negative curvature or positive edges, we streamline the connectivity of the graph. In this scenario, the

value of $W_1(\mu_i, \mu_j)$ in Equation 3 decreases, effectively reducing the count of strongly negatively curved edges (Topping et al., 2022) that often act as bottlenecks.

Furthermore, sparsifying the graph offers several advantages: (1) it aids in mitigating oversmoothing concerns (Rong et al., 2019), (2) significantly decreases the graph's diameter, thus reducing the number of hops required to connect two nodes and preventing over-squashing issues (Deac et al., 2022). Our empirical findings demonstrate that employing a one-hop strategy proves beneficial in constraining bottlenecks, as evidenced by an increase in the normalized spectral gap following the rewiring process.

Two-hop curvature. Leveraging a multi-hop neighborhood addresses a limitation in classical Message Passing Neural Networks (MPNNs), where nodes can only communicate directly with their neighbors. By expanding the graph connectivity through multiple hops, information transmission becomes possible even with nodes that are distant (Brüel-Gabrielsson et al., 2022)(Abboud et al., 2022). This approach eliminates the need to iterate messages across powers of the adjacency matrix, mitigating the risk of over-squashing (Topping et al., 2022).

However, depending on the graph's size, this may significantly escalate the computational cost of the method proposed by Gutteridge et al. (Gutteridge et al., 2023). By focusing solely on a two-hop neighborhood based on a specific curvature of the edges, it becomes feasible to restrict the graph's densification. This not only reduces the computational burden of the two-hop approach but also facilitates efficient information exchange between distant nodes.

Utilizing one-hop and two-hop curvature across layers. It's essential to bear in mind that when the distance k between nodes i and j surpasses one, their interaction is confined to the k^{th} layer.

In the realm of two-layer Graph Neural Networks (GNNs), the integration of one-hop and subsequently two-hop curvature between layers expedites interactions between distant nodes. This strategy restricts dynamic rewiring message passing (Gutteridge et al., 2023) to either positive or negative curvatures, as demonstrated in (Gutteridge et al., 2023), effectively mitigating concerns related to over-squashing.

This tailored framework not only accelerates information exchange but also addresses a key limitation of the paper—its exclusive suitability for very deep GNN models—by imposing constraints on the dynamic rewiring process.

4. Experiments

We carried out experiments on the eleven different node classification datasets for which we presented the homophily measures in Table 1 and compared the results obtained by our proposed method to four other methods based on rewiring techniques.

4.1. Datasets

We provide a detailed overview of the datasets used in our study, which include 7 heterophilic datasets and 4 homophilic datasets. The current datasets vary in size and have very different structures.

- **WebKB**: This dataset consists of pages from Cornell, Texas, and Wisconsin, where nodes represent web pages and edges denote hyperlinks between them. Node features are represented using a bag-of-words model based on the content of the web pages. The classification labels include student, project, course, staff, and faculty.

- **Actor**: In this dataset, each node represents an actor, and an edge between two nodes indicates their co-occurrence on the same Wikipedia page (Tang et al., 2009). Node features are derived from keywords found in the corresponding Wikipedia pages. The goal is to classify the nodes into five distinct categories.

- **Wikipedia Network**: The Squirrel and Chameleon datasets are included in this category, where nodes represent web pages and edges indicate mutual links between these pages (Rozemberczki et al., 2021). Node features correspond to informative nouns extracted from the Wikipedia pages. The nodes are categorized into five classes based on their popularity, defined by the average monthly traffic of each web page.

- **Roman Empire**: This dataset is based on the article about the Roman Empire from English Wikipedia, which is one of the longest articles available (Platonov et al., 2023). Each node corresponds to a word in the text. An edge connects two words if they either follow each other in the text or are linked in the dependency tree of a sentence. The classification for each node is determined by its syntactic role.

- **Scientific Publication Networks**: The Cora and Citeseer datasets (Sen et al., 2008) describe citations among scientific publications. Each publication is represented by a binary bag-of-words model indicating the presence or absence of specific words in the publication's abstract. The classes correspond to the categories of the publications.

- **Amazon**: This dataset includes two subsets, Amazon Computers and Amazon Photo, derived from the Amazon purchase graph (McAuley et al., 2015). Nodes represent products, while edges indicate whether two products are frequently purchased together. The features are the bags of words from the product descriptions, and the classes represent different product categories.

The statistics for these datasets are summarized in Table 2. Additionally, we present the construction time of the curvature matrix using both Ollivier (O) and Augmented Forman (AF) methods (measured in seconds).

Scalability. Our method, based on Augmented Forman Curvature, demonstrates superior scalability compared to most graph rewiring techniques. The computational complexity of Balanced Forman Curvature is $\mathcal{O}(|\mathcal{E}|d_{\max}^2)$, where $d_{\max}$ is the maximum node degree, making it significantly more resource-intensive than both Ollivier and Augmented Forman curvature. As a result, the Balanced Forman curvature calculations required GPU acceleration to achieve practical efficiency.

Our results show that Augmented Forman Curvature is far more computationally efficient than Ollivier Curvature, especially on larger graphs. For instance, on the penn94 dataset (Lim et al., 2021) (with 41,554 nodes and 1,362,229 edges), the Augmented Forman Curvature was computed in just 675 seconds, while Ollivier Curvature ran out of memory (OOM) with 12GB of RAM

4.2. Baseline

We use the two most popular GNNs, GCN (Kipf and Welling, 2017) and GAT (Veličković et al., 2018) as a basis and we compare our method with four other methods based on structural rewiring techniques (Attali et al., 2024). We provide the results of FA (Alon and Yahav, 2021), DIGL (Klicpera et al.,

Dataset	# Nodes	# Edges	# C	O-time	AF- time
Squirrel	5021	217073	5	$\approx$ 836	$\approx$202
Chameleon	2277	36101	5	$\approx$ 11	$\approx$9
Texas	181	309	5	$\approx$ 1	$\approx$1
Wisconsin	251	499	5	$\approx$ 1	$\approx$1
Cornell	181	295	5	$\approx$ 1	$\approx$1
R-empire	22662	32927	18	$\approx$ 75	$\approx$2
Actor	7600	33544	5	$\approx$ 10	$\approx$ 4
Cora	2 708	5 429	7	$\approx$ 2	$\approx$ 1
Citeseer	3 312	4 715	6	$\approx$ 3	$\approx$ 1
Computers	13752	245861	10	$\approx$ 287	$\approx$ 135
Photo	7650	119081	8	$\approx$ 61	$\approx$34

Table 2: Number of nodes (# Nodes), edges (# Edges) and node labels (#C) for each dataset. We also show the computation time (in seconds) for Ollivier (O-time) and Augmented Forman (AF-time) curvatures on the graphs.

2019)[2], SDRF(Topping et al., 2022)[3], FOSR(Karhadkar et al., 2023)[4] and BORF (Nguyen et al., 2023). [5].
For SDRF, we use the hyperparameters that have been defined in the original publication and fine-tune the number of iterations. For FOSR, we fine-tuned the number of iterations. For BORF, we fine-tuned the top values from the sets of the number of batches n $\{1, 2, 3\}$, the number of edges added per batch h $\{10, 20, 30, 40\}$, and the number of edges removed per batch k $\{10, 20, 30, 40\}$. For DIGL we fine tune top k for $\{8, 16, 32, 64, 128\}$ and $\{0.05, 0.1, 0.15\}$ for the personalized PageRank (Page et al., 1998).

4.3. Setup

For the experiments we use the same framework as (Pei et al., 2020) ,(Attali et al., 2024) to evaluate the robustness of each method. Thus, we fix the number of layers to 2, the dropout to $= 0.5$, learning rate to 0.005, patience of 100 epochs, weight decay of $5E^{-6}$ (Texas, Wisconsin and Cornell) or $5E^{-5}$ (other datasets). The number of hidden states is 32 (Texas, Wisconsin and

[2]https://github.com/gasteigerjo/gdc
[3]https://github.com/jctops/understanding-oversquashing/tree/main
[4]https://github.com/kedar2/FoSR/tree/main
[5]https://github.com/hieubkvn123/revisiting-gnn-curvature

Cornell), 48 (Squirrel, Chameleon and Roman-Empire), 32 (Actor) and 16 (Cora and Citeseer) except for Amazon Photo and Computers where the hidden states is 64 and we use a learning rate of 0.01 following the usual framework presented in (Shchur et al., 2018).

For all the graphs datasets we take a random sample of nodes of 60% for training, and 20% for validation and 20% for testing. We report the average accuracy of each method on 100 random samples.

Curvature-Constrained Message Passing (CCMP). We first calculate the average curvature over all graph edges $\mu = \frac{\sum_{\{i,j|A_{ij}=1\}} c_{ij}}{|\mathcal{E}|}$. Then, we can split the adjacency matrix A into two adjacency matrices A^+ and A^- such as: $A^+_{ij} = 1 \;\; if \;\; c_{ij} \geq \mu$ and $A^-_{ij} = 1 \;\; if \;\; c_{ij} \leq \mu$. For each dataset, we then choose to propagate the messages either along the positive or negative edges (i.e; we choose A^+ or A^- as adjacency matrix), depending on which one presents the highest gain compared to the average homophily of the original graph (as it can be seen in Table A.9). We also choose a 1-hop or a 2-hop strategy depending on the size of the graph. 2-hops strategies are well adapted to the small datasets, such as Cornell, Wisconsin and Texas; therefore, we use 2-hop in this case.

We indicate the Olliver curvature with CCMP_O, the augmented Forman curvature with CCMP_{AF} and the Balanced Forman with CCMP_{BF}.

Method	Cora	Citeseer	Photo	Computers
Base(GCN)	87.73 ±0.25	76.01 ±0.25	89.89 ±0.37	80.45 ±0.56
DIGL	88.22 ±0.28	76.18 ±0.34	90.31 ±0.43	83.04±0.43
FA	29.86 ±0.28	22.31 ±0.34	OOM	OOM
SDRF	87.73 ±0.31	76.43±0.32	$\geq 24H$	$\geq 24H$
FOSR	87.94±0.26	76.34 ±0.27	90.24 ±0.31	80.78 ±0.43
BORF	87.80 ±0.26	76.49 ±0.28	$\geq 24H$	$\geq 24H$
CCMP$_0$	85.34 ±0.29	75.53±0.29	90.30 ±0.35	82.40 ±0.43
CCMP$_{AF}$	85.60 ±0.37	75.76 ±0.39	90.31 ±0.38	81.84 ±0.45
CCMP$_{BF}$	86.01±0.32	75.79 ±0.41	89.31 ±0.32	82.67 ±0.44

Table 3: Node classification results on **homophilic** datasets with **GCN** as backbone. The top three accuracy are coloured as First, Second and Third, respectively.

4.4. Results

Tables 5, 6, 3 and 4 present the results of our experiments.

Method	Cora	Citeseer	Photo	Computers
Base(GAT)	87.65 ±0.24	76.08 ±0.27	88.76 ±0.39	80.72 ±0.53
DIGL	88.31 ±0.29	76.22 ±0.34	90.32±0.46	83.28 ±0.49
FA	30.44 ±0.26	23.11 ±0.32	OOM	OOM
SDRF	88.11 ±0.28	76.16 ±0.31	$\geq 24H$	$\geq 24H$
FOSR	88.13 ±0.27	75.94±0.32	90.12 ±0.41	80.78 ±0.51
BORF	87.72±0.27	76.44 ±0.33	$\geq 24H$	$\geq 24H$
CCMP$_O$	84.82 ±0.28	75.82±0.30	89.57 ±0.33	81.97 ±0.47
CCMP$_{AF}$	86.16 ±0.32	76.18 ±0.44	89.88 ±0.22	81.96 ±0.51
CCMP$_{BF}$	87.39 ±0.34	75.79 ±0.34	89.21 ±0.29	81.98 ±0.49

Table 4: Node classification results on **homophilic** datasets with **GAT** as backbone. The top three accuracy are coloured as First, Second and Third, respectively.

	Chameleon	Squirrel	Actor	Texas	Wisconsin	Cornell	R-Empire
Base (GCN)	65.35±0.54	51.30±0.38	30.02±0.22	56.19 ±1.61	55.12±1.51	44.78 ±1.45	51.66 ±0.17
DIGL	54.82 ±0.48	40.53 ±0.29	26.75 ±0.23	45.95 ±1.58	46.90 ±1.28	44.46 ±1.37	53.93 ±0.14
FA	26.34 ±0.61	22.88 ±0.42	26.03±0.30	55.93 ±1.76	46.77±1.48	45.33±1.55	OOM
SRDF	63.08 ±0.37	49.11±0.28	31.85 ±0.22	59.79 ±1.71	58.49 ±1.23	47.73 ±1.51	52.53 ±0.13
FOSR	67.98 ±0.40	52.63 ±0.30	29.26±0.23	61.35 ±1.25	55.60 ±1.25	45.11 ±1.47	52.38 ±0.21
BORF	65.35 ±0.51	$\geq$ 24h	31.36±0.27	56.30±1.45	55.37±1.47	46.81±1.56	52.94 ±0.21
CCMP$_O$	61.22 ±0.45	53.34 ±0.33	32.55±0.22	69.38±1.81	66.04±1.31	58.91 ±1.82	58.14 ±0.17
CCMP$_{AF}$	65.66 ±0.44	54.79±0.33	34.59±0.24	69.67 ±1.64	67.80 ±1.49	58.95 ±1.63	58.91 ±0.19
CCMP$_{BF}$	62.29 ±0.49	53.04 ±0.41	32.90±0.28	68.59 ±1.99	64.37 ±1.42	59.41 ±1.57)	58.13 ±0.15

Table 5: Node classification results on **heterophilic** datasets with **GCN** as backbone. The top three accuracy results are coloured as First, Second and Third, respectively.

	Chameleon	Squirrel	Actor	Texas	Wisconsin	Cornell	R-Empire
Base (GAT)	65.07 ±0.41	50.87 ±0.56	29.92 ±0.23	56.84 ±1.61	53.58 ±1.39	46.05 ±1.49	49.23 ±0.33
DIGL	56.34 ±0.43	41.65 ±0.68	31.22 ±0.47	46.49 ±1.63	46.29 ±1.47	44.05 ±1.44	53.89 ±0.16
FA	27.11 ±0.56	21.49 ±0.71	28.20 ±0.51	56.17 ±1.71	46.95 ±1.52	44.60 ±1.74	OOM
SRDF	63.15 ±0.44	50.36 ± 0.38	31.47 ±0.25	57.45 ±1.62	56.80 ±1.29	48.03 ±1.66	50.75 ±0.17
FOSR	66.61 ±0.45	52.02 ±0.43	29.73 ±0.24	61.85 ±1.41	54.06 ±1.27	48.30 ±1.61	49.54 ±0.31
BORF	66.92 ±0.60	$\geq$ 24h	29.64 ±0.33	56.68 ±1.49	55.39 ±1.23	48.57 ±1.56	51.03 ±0.26
CCMP$_O$	62.86 ±0.52	52.69 ±0.34	32.32 ±0.27	73.81 ±1.29	65.71±1.23	60.03 ±1.41	56.83 ±0.19
CCMP$_{AF}$	65.59 ±0.43	54.74 ±0.52	34.23 ±0.23	70.65 ±1.36	68.59 ±1.41	59.81 ±1.49	56.78 ±0.39
CCMP$_{BF}$	64.91 ±0.54	51.67 ±0.48	32.28 ±0.25	72.73 ±1.45	67.71 ±1.22	61.01 ±1.61	57.10 ±0.22

Table 6: Node classification results on **heterophilic** datasets with **GAT** as backbone. The top three results are coloured as First, Second and Third, respectively.

Homophilic datasets. Table 3 and Table 4 show that the rewiring strategies aimed at mitigating over-squashing do not improve significantly the results on homophilic dataset, in comparison to the GCN and GAT baselines. Given the characteristics of the graphs, this is an expected result. DIGL, on the other hand, obtains the best results on three datasets over four. The reason is that DIGL improves connectivity among nodes with short diffusion paths. This rewiring process allows to add positively curved edges that improve the connectivity of nodes that share the same label according to the value of β^+ in Table A.9.

Heterophilic datasets. As shown in Tables 5 and 6, our method (CCMP) achieves the best results for six out of seven heterophilic datasets. CCMP_O, CCMP_{AF}, CCMP_{BF} on average, exceed the results obtained on the original adjacency matrix with a basic GCN and GAT by 14.24%, 16.55%, and 13,92% respectively. It can be noted that this result holds no matter which curvature is chosen for our CCMP method. Therefore, AF+CCMP seems the best compromise considering the computational costs (see Table 7) and the overall performance: it is the best method in 3 datasets over 7, and among the top 3 choices for the other 4.
Regarding the other methods, as expected on heterophilic datasets the SDRF, FOSR and BORF methods improve performance compared to the baseline and also outperform DIGL due to the tendency of neighboring nodes to have different labels.

Computational Costs. Utilizing a one-hop curvature allows the reduction of the graph size. Table 7 shows the average run execution time using CCMP and the base GCN. The results show an important decrease in execution time in particular for the larger graphs.

	CCMP_O	CCMP_{AF}	CCMP_{BF}
Chameleon	**-29.8%**	**-31.4%**	**-26.2%**
Squirrel	**-24.2%**	**-19.2%**	**-2.8%**
Actor	**-6.2%**	8.2%	1.6%
Roman-Empire	**-37.9%**	**-10.3%**	**-27.7%**
Cora	**-5.6%**	**-10.1%**	**-8.2%**
Citeseer	**-6.7%**	**-0.7%**	**-4.4%**
Photo	**-23.1%**	**-15.8%**	**-31.6%**
Computers	**-28.8%**	**-4.8%**	**-28.9%**

Table 7: Comparison of the execution time for the average of the runs according to the datasets using CCMP and the base GCN. In bold where the execution time is lower than the original method.

Properties of the rewired graph. To show that our approach does indeed mitigate over-squashing, we present the evolution of the normalized spectral gap compared to the original graph in Table 8. In general, regardless of the curvature used we observe that the spectral gap of rewired graph is higher than that of the original graph. Consequently, the resulting graph has fewer bottleneck structures, thereby enhancing the overall quality of signal propagation across nodes.

	O +	O -	AF +	AF -	BF +	BF -
Cham.	**46.7%**	18.8%	**16.2%**	63.1%	63.8%	**23.2%**
Squi.	**58.4%**	6.6%	1.0	**42.8%**	72.3%	**9.7%**
Actor	31.0%	**56.6%**	5.7	**384.1%**	25.1%	**74.2%**
Texas	-23.6%	**-1.0%**	-14.3%	**-19.7%**	-21.8%	**-5.3%**
Wisc.	7.0%	**37.4%**	-22.6%	**-5.1%**	-7.7%	**-14.8%**
Cornell	-18.9%	**12.0%**	-7.4%	**23.5%**	-17.21%	**9.6%**
R-Emp	34.1%	**86.2%**	103	**15.4%**	53.2%	**38.2%**

Table 8: Evolution of the normalised spectral gap as a function of curvature. In bold, the curvature used in our experiments.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we introduce a methodology applicable to any Message Passing Neural Network (MPNN) architecture, enabling the distribution of messages based on the curvature of the graph's edges. This innovative approach addresses over-squashing, a prominent limitation in classical MPNN architectures. By incorporating a novel curvature-constrained homophily metric, we have purpose various method variants for propagating information along curved edges, encompassing propagation along negative or positive edges, with either one or two hops. Our experiments demonstrate a substantial enhancement compared to equivalent state-of-the-art methods that utilize rewiring, empirically validating the efficacy of curvature-based constraints in message passing for bottleneck reduction.

In future research, we want to study the effect of using very deep GNN models with different curvature adjacency matrices for long-range graph benchmarks. We also plan to create a new attention mechanism by taking into account not only the features of the nodes but also the local structure of the graph via the curvature of the edges of the graph.

References

Ralph Abboud, Radoslav Dimitrov, and Ismail Ilkan Ceylan. 2022. Shortest path networks for graph property prediction. In *Learning on Graphs Conference*. PMLR, 5–1.

Noga Alon and Vitali D Milman. 1984. Eigenvalues, expanders and superconcentrators. In *25th Annual Symposium onFoundations of Computer Science, 1984*. IEEE, 320–322.

Uri Alon and Eran Yahav. 2021. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.

Hugo Attali, Davide Buscaldi, and Nathalie Pernelle. 2024. Delaunay Graph: Addressing Over-Squashing and Over-Smoothing Using Delaunay Triangulation. In *Forty-first International Conference on Machine Learning*. `https://openreview.net/forum?id=uyhjKoaIQa`

Pradeep Kr Banerjee, Kedar Karhadkar, Yu Guang Wang, Uri Alon, and Guido Montúfar. 2022. Oversquashing in GNNs through the lens of information contraction and graph expansion. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 1–8.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018).

Rickard Brüel-Gabrielsson, Mikhail Yurochkin, and Justin Solomon. 2022. Rewiring with positional encodings for graph neural networks. *arXiv preprint arXiv:2201.12674* (2022).

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs.

Chen Cai and Yusu Wang. 2020. A note on over-smoothing for graph neural networks. *Graph Representation Learning* (2020).

Jeff Cheeger. 1970. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in Analysis: A Symposium in Honor of Salomon Bochner (PMS-31)*. Princeton University Press.

Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. American Mathematical Soc.

Andreea Deac, Marc Lackenby, and Petar Veličković. 2022. Expander graph propagation. In *Learning on Graphs Conference*. PMLR, 38–1.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. 2020. A fair comparison of graph neural networks for graph classification. In *ICLR*.

Robin Forman. 2003. Bochner's Method for Cell Complexes and Combinatorial Ricci Curvature.

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

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, Vol. 1. IEEE, 347–352.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 2. IEEE, 729–734.

Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. 2023. DRew: Dynamically Rewired Message Passing with Delay. In *International Conference on Machine Learning*. PMLR, 12252–12267.

Jürgen Jost and Shiping Liu. 2014. Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete & Computational Geometry* 51, 2 (2014), 300–322.

Kedar Karhadkar, Pradeep Kr Banerjee, and Guido Montúfar. 2023. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *International Conference on Learning Representations (ICLR)*.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. In *Advances in neural information processing systems (NeurIPS)*.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in neural information processing systems (NeurIPS)*. 20887–20902.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. 2023. Revisiting over-smoothing and over-squashing using ollivier-ricci curvature. In *International Conference on Machine Learning*. PMLR, 25956–25979.

Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. 2015. Ricci curvature of the Internet topology. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2758–2766. `https://doi.org/10.1109/INFOCOM.2015.7218668`

Chien-Chun Ni, Yu-Yao Lin, Jie Gao, and Xianfeng Gu. 2018. Network alignment by discrete Ollivier-Ricci flow. In *International Symposium on Graph Drawing and Network Visualization*. Springer, 447–462.

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. 2019. Community detection on networks with Ricci flow. *Scientific reports* 9, 1 (2019), 9984.

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

Kenta Oono and Taiji Suzuki. 2020. Graph neural networks exponentially lose expressive power for node classification. *Proceedings of the International Conference on Learning Representations* (2020).

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. *The pagerank citation ranking: Bring order to the web*. Technical Report. Technical report, stanford University.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. In *Advances in neural information processing systems (ICLR)*.

Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. 2022. Characterizing graph datasets for node classification: Beyond homophily-heterophily dichotomy. *arXiv preprint arXiv:2209.06177* (2022).

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: are we really making progress?. In *International Conference on Learning Representations*.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. In *International conference on learning representations (ICLR)*.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

Areejit Samal, RP Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost. 2018. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Scientific reports* 8, 1 (2018), 8650.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. , 61–80 pages.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

Jayson Sia, Edmond Jonckheere, and Paul Bogdan. 2019. Ollivier-ricci curvature-based method to community detection in complex networks. *Scientific reports* 9, 1 (2019), 9800.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 807–816.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. *Proceedings of the International Conference on Learning Representations* (2022).

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *Proceedings of the International Conference on Learning Representations*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. In *IEEE transactions on neural networks and learning systems*, Vol. 32. IEEE, 4–24.

Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. 2022. Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications* 33 (2022), 1–19.

Ze Ye, Kin Sum Liu, Tengfei Ma, Jie Gao, and Chao Chen. 2019. Curvature graph network. In *International conference on learning representations*.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in neural information processing systems*, Vol. 31.

Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082* (2022).

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems* 33 (2020), 7793–7804.

Appendix A. Appendices

Appendix A.1. Details of configuration for CCMP

- The optimal configurations for CCMP_O is:

  (1) for Cora, Citeseer,Amazon photo, Amazon Computers and Chameleon a one-hop positively curved adjacency matrix for both layers. (2) for Texas, Wisconsin and Cornell, a two-hop negatively curved adjacency matrix for both layers. (3) for Squirrel, Actor, Roman-Empire datasets, a negatively curved one-hop adjacency matrix on two layers,

- The optimal configurations for CCMP_{AF} is:

  (1) for Cora, Citeseer, Amazon photo, Amazon Computers, Chameleon and Squirrel a one-hop positively curved adjacency matrix for both layers. (2) for Texas, Wisconsin and Cornell, a two-hop negatively curved adjacency matrix for both layers. (3) For Actor and Romain-empire we use a negatively curved one-hop adjacency matrix on two layers.

- The optimal configurations for CCMP_{BF} is:

  (1) for Cora, Citeseer, Amazon photo and Amazon Computers a one-hop positively curved adjacency matrix for both layers. (2) for Texas, Wisconsin and Cornell, a two-hop negatively curved adjacency matrix for both layers. (3) For Chameleon, Squirrel, Actor and Romain-empire we use a negatively curved one-hop adjacency matrix on two layers,

Appendix A.2. Homophily of datasets according to curvature

Here we specify the details of the Curvature-Constrained homophily used in the experiments for our rewiring methods for layers 1/2.

	Dataset	β	β-CCMP$_O$	β-CCMP$_{AF}$	β-CCMP$_{BF}$
Heterophilic	Squirrel	0.23/0.23	0.28/0.28	0.24/0.24	0.24/0.24
	Chameleon	0.26/0.26	0.29/0.29	0.28/0.28	0.31/0.31
	Texas	0.31/0.31	0.47/0.47	0.56/0.56	0.47/0.47
	Wisconsin	0.36/0.36	0.38/0.38	0.42/0.42	0.38/0.38
	Cornell	0.34/0.34	0.40/0.40	0.39/0.39	0.41/0.41
	R-empire/	0.29/0.29	0.48/0.48	0.33/0.33	0.41/O.41
	Actor	0.32/0.32	0.73/0.73	0.64/0.64	0.36/0.36
Homophilic	Cora	0.84/0.84	0.95/0.95	0.98/0.98	0.93/0.93
	Citeseer	0.81/0.81	0.86/0.86	0.89/0.89	0.83/0.83
	Photo	0.84/0.84	0.94/0.94	0.89/0.89	0.92/0.92
	Computers	0.78/0.78	0.93/0.93	0.83/0.83	0.86/0.86

Table A.9: Comparison of edge homophily measures. The last column reports the max gain in homophily obtained by using the curvature-constrained edge homophily as opposed to edge homophily.