CONGREGATE: Contrastive Graph Clustering in Curvature Spaces

Li Sun[1*], **Feiyang Wang**[2], **Junda Ye**[2], **Hao Peng**[3*], **Philip S. Yu**[4]
[1]North China Electric Power University, Beijing 102206, China
[2]Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]Beihang University, Beijing 100191, China
[4]Department of Computer Science, University of Illinois at Chicago, IL, USA
ccesunli@ncepu.edu.cn; {fywang, jundaye}@bupt.edu.cn; penghao@buaa.edu.cn; psyu@uic.edu

Abstract

Graph clustering is a longstanding research topic, and has achieved remarkable success with the deep learning methods in recent years. Nevertheless, we observe that several important issues largely remain open. On the one hand, graph clustering from the geometric perspective is appealing but has rarely been touched before, as it lacks a promising space for geometric clustering. On the other hand, contrastive learning boosts the deep graph clustering but usually struggles in either graph augmentation or hard sample mining. To bridge this gap, we rethink the problem of graph clustering from geometric perspective and, to the best of our knowledge, make the first attempt to *introduce a heterogeneous curvature space to graph clustering problem*. Correspondingly, we present a novel end-to-end contrastive graph clustering model named CONGREGATE, addressing geometric graph clustering with Ricci curvatures. To support geometric clustering, we construct a theoretically grounded Heterogeneous Curvature Space where deep representations are generated via the product of the proposed *fully Riemannian* graph convolutional nets. Thereafter, we train the graph clusters by an *augmentation-free* reweighted contrastive approach where we pay more attention to *both hard negatives and hard positives* in our curvature space. Empirical results on real-world graphs show that our model outperforms the state-of-the-art competitors.

1 Introduction

Graph clustering aims to group nodes into different clusters so that the intra-cluster nodes share higher similarity than the inter-cluster ones, receiving continuous research attention [Yin *et al.*, 2017]. The state-of-the-art clustering performance on graphs has been achieved by deep clustering methods in recent years [Liu *et al.*, 2023; Wang *et al.*, 2019; Li *et al.*, 2020]. Meanwhile, we find that several important issues on deep graph clustering still largely remain open.

The first issue is on the geometric graph clustering. In the literature, classic concepts such as modularity [Li *et al.*, 2022], conductance [Duval and Malliaros, 2022] and motifs [Jia *et al.*, 2019] are frequently revisited. Little effort has been devoted to clustering from a geometric perspective. In the Riemannian geometry, **Ricci curvatures** on the edges can help determine the cluster boundary [Jost and Liu, 2014], thereby showing the density and clustering behavior among the nodes. However, graph clustering has rarely been touched yet in Riemannian geometry, since *it lacks a promising Riemannian space for graph clustering*. Most existing graph representation spaces present a single curvature radius, independent of nodes/edges [Xiong *et al.*, 2022b; Chami *et al.*, 2019; Law, 2021], and cannot allow for a closer look over the various curvatures for graph clustering. Also, typical clustering algorithms in Euclidean space (e.g., K-means) cannot be directly applied as an alternative, due to the inherent difference in geometry. Consequently, it calls for a new Riemannian curvature space, *supporting a fine-grained curvature modeling for geometric clustering*.

The second is on the unsupervised learning. Deep models are typically trained by the supervisions while graph clustering is unsupervised by nature. Recently, the contrastive clustering without external supervision draws dramatic attention [Park *et al.*, 2022; Devvrit *et al.*, 2022; Li *et al.*, 2022]. *In the line of contrastive graph clustering, the issues of augmentation and hard samples are still unclear in general.* Unlike the easily obtained augmentations on images, graph augmentation is nontrivial [Hassani and Ahmadi, 2020]. In addition, the noise injected in this process usually requires a careful treatment to avoid misleading on graph clustering [Gong *et al.*, 2022]. Robinson *et al.* [2021] point out the hardness unawareness of typical loss function such as InfoNCE. Hard negative samples have shown to be effective for graph contrastive learning [Xia *et al.*, 2022], but little effort is made to its counterpart, *hard positive samples*. In fact, the hard positives in our context are the border nodes of a cluster, and plays a crucial role in clustering performance. Unfortunately, hard sample mining in curvature space largely remains open.

Motivated by the observations above, we rethink the problem of graph clustering from the geometric perspective, and make the first attempt to address graph clustering in a novel **Curvature Space**, *rather than traditional single curvature ones*, with an advanced contrastive loss.

To this end, we propose a novel end-to-end contrastive graph clustering model in curvature spaces (CONGREGATE), where we approach graph clustering via geometric clustering

with Ricci curvatures so that positive Ricci curvature groups the nodes while negative Ricci departs them in spirit of the famous Ricci flow. To address the fine-grained curvature modeling for graph clustering (*the first issue*), we introduce a novel *Heterogeneous Curvature Space*, which is a key innovation of our work. It is designed as the product of learnable factor manifolds and multiple free coordinates. We prove that the proposed space allows for different curvatures on different regions, and the fine-grained node curvatures can be inferred to accomplish curvature modeling. Accordingly, we generate deep representations via the product of Graph Convolutional Nets (GCNs), where *fully Riemannian* GCN is designed to address the inferior caused by tangent spaces. To address the unsupervised learning (*the second issue*), we propose a rewighted geometric contrastive approach in our curvature space. On the one hand, our approach is *free of augmentation* as we contrast across the geometric views generated from the proposed heterogeneous curvature space itself. On the other hand, we equip a novel dual reweighting to the Node-to-Node and Node-to-Cluster contrastive losses to train the clusters. In this way, we pay more attention to *both hard negatives and hard positives* when maximizing intra-cluster similarity and minimizing inter-cluster similarity.

To sum up, the noteworthy contributions are listed below:

- *Problem*. We rethink the graph clustering from geometric respective. To the best of our knowledge, we are the first to introduce the heterogeneous curvature space, supporting fine-grained curvatures modeling, to the problem of graph clustering.
- *Methodology*. We propose an end-to-end CONGREGATE free of graph augmentation, in which we approach geometric graph clustering with the reweighting contrastive loss in the proposed *heterogeneous curvature space*, paying attention to hard positives and hard negatives.
- *Experiments*. We evaluate the superiority of our model with 19 strong competitors on 4 datasets, examine the proposed components by ablation study, and further discuss why Ricci curvature works.

2 Preliminaries

In this section, we first introduce the necessary fundamentals of Riemannian geometry for better understanding our work, and then formulate the studied problem in this paper. In short, *we are interested in the end-to-end graph clustering in a novel curvature space*.

2.1 Riemannian Geometry

Manifold. A Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold $\mathcal{M}$ endowed with a Riemannian metric g. Every point $x \in \mathcal{M}$ is associated with a Euclidean-like *tangent space* $\mathcal{T}_x\mathcal{M}$ on which the metric g is defined. The *exponential map* projects from the tangent space onto the manifold, and the *logarithmic map* does inversely [Lee, 2013].

Curvature. For each point x in the manifold, it is coupled with a curvature c_x describing how the space around x derives from being flat and a corresponding curvature radius $\frac{1}{|c_x|}$. When c_x is equal everywhere in the manifold, it induces a **homogeneous** curvature space (a.k.a. constant curvature space) with a simplified notation of scalar curvature c. Concretely, it is said to be *hyperbolic* $\mathbb{H}$ if $c < 0$, and *hyperspherical* $\mathbb{S}$ if $c > 0$. *Euclidean* space $\mathbb{R}$ is special case with $c = 0$. On the contrary, **heterogeneous** curvature space refers to a manifold whose curvatures on different regions are not the same, which is a more practical yet challenging case.

2.2 Problem Formulation

In this paper, we consider the node clustering on attributed graphs. An attributed graph is described as a triplet of $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ is the set of N nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set, and $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the attribute matrix. Let K denote the number of node clusters. The node-to-cluster assignment is described as the *cluster membership* vector $\boldsymbol{\pi}_i \in \mathbb{R}^K$ attached to node v_i. $\boldsymbol{\pi}_i$ is a stochastic vector adding up to 1, whose k-th element π_{ik} is the probability of v_i belonging to cluster k. Now, we formulate the problem of Geometric Graph Clustering in Generic Curvature Space.

Problem Definition. *Given $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, the goal of our problem is to learn an encoder $f : v_i \rightarrow [\boldsymbol{z}_i, \boldsymbol{\pi}_i], \forall v \in \mathcal{V}$ that 1) directly outputs cluster membership $\boldsymbol{\pi}_i$* (end-to-end) *so that the nodes are more similar to those grouped in the same cluster than the nodes in different clusters and 2) the node encodings in the generic curvature space $\boldsymbol{z}_i \in \mathcal{M}$,* supporting the geometric graph clustering.

Distinguishing with the prior works, we rethink the problem of graph clustering from the geometric perspective, and make the first attempt to study graph clustering in a novel *Curvature Space*, rather than traditional single curvature ones.

Notations. The lowercase x, boldfaced $\boldsymbol{x}$ and uppercase $\mathbf{X}$ denote scalar, vector and matrix, respectively.

3 Methodology: CONGREGATE

We propose an end-to-end contrastive graph clustering model (CONGREGATE) where *we introduce the first curvature space to graph clustering*, a key innovation of our work. In brief, we directly learn the node clusters by training randomly initialized centroids $\{\boldsymbol{\phi}_k\}_{k=1,\cdots,K}$ in a novel curvature space. $\boldsymbol{\phi}_k$ is the centroid of cluster k. The soft assignment of node v_i to cluster k is given as $\pi_{ik} = Normalize(\delta(\boldsymbol{z}_i, \boldsymbol{\phi}_k))$, where the similarity $\delta(\boldsymbol{z}_i, \boldsymbol{\phi}_k) = exp(-d_{\mathcal{P}}(\boldsymbol{z}_i, \boldsymbol{\phi}_k))$ and $d_{\mathcal{P}}$ is distance metric in our curvature space. Softmax normalization is applied so that $\boldsymbol{\pi}_i$ adds up to 1.

We illustrate our model in Figure 1. Concretely, we present a geometric clustering approach with Ricci curvatures (**Sec 3.1**), introduce the novel heterogeneous curvature space (**Sec 3.2**), and train cluster centroids by the proposed reweighted contrastive loss in our curvature space (**Sec 3.3**).

3.1 Geometric Clustering with Ricci Curvature

In CONGREGATE, we address graph clustering from a geometric perspective, more concretely, *the notion of Ricci curvature*, and formulate a novel geometric clustering loss.

We first discuss why Ricci curvature clusters nodes. Let us begin with its definition [Jost and Liu, 2014; Lin *et al.*, 2011]:

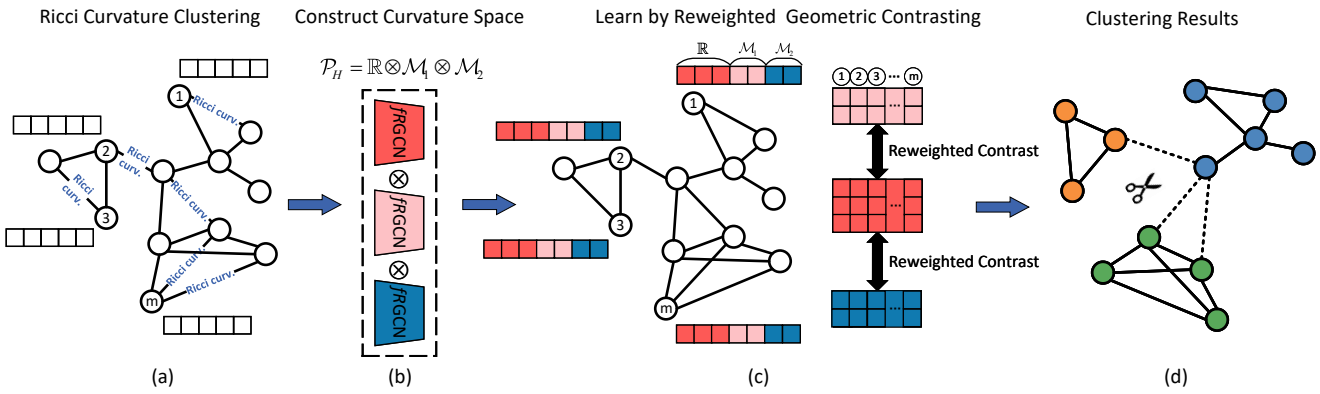


Figure 1: Illustration of CONGREGATE. (a) We address graph clustering from geometric perspective with Ricci curvatures. (b) We construct a novel curvature space where we generate deep representations via the product of proposed ***f*RGCN**s. (c) Our model is trained by a reweighted contrastive loss across geometric views (red/magenta/blue) free of augmentation. (d) We obtain clustering results in an end-to-end fashion.

Given a graph with mass distribution $m_i^\lambda(\cdot)$ on v_i's neighbor nodes, Ricci curvature $Ric(i,j)$ of edge (v_i, v_j) is defined as

$$Ric(i,j) = 1 - \frac{W(m_i^\lambda, m_j^\lambda)}{d_G(v_i, v_j)}, \tag{1}$$

and $W(m_i^\lambda, m_j^\lambda)$ is the Wasserstein distance between the mass distributions on nodes, where $m_i^\lambda(\cdot)$ is defined as

$$m_i^\lambda(v_j) = \begin{cases} \lambda & \text{if } v_j = v_i \\ \frac{1-\lambda}{degree_i} & \text{if } v_j \in \mathcal{N}_i, \end{cases} \tag{2}$$

where d_G is the length of shortest path on the graph, and λ is a control parameter. The intuition is that *the Ricci curvature of an edge describes the overlap extent between neighborhoods of its two end nodes, and thus signifies the density among nodes.* Specifically, if v_i and v_j belong to different clusters, it is costing to move the distribution m_i^λ to m_j^λ due to fewer common neighbors. The less overlapped neighborhoods present large $W(m_i^\lambda, m_j^\lambda)$ and negative $Ric(i,j)$. On the contrary, intra-cluster edges are most positively curved, and the nodes within the cluster are densely connected.

With the observation above, we connect the Ricci curvature on edges to the density among the nodes. Then, intra-cluster density is formulated as summing the $Ric(i,j)$ whose end nodes belong to the same cluster,

$$D_{intra} = \frac{1}{|\mathcal{E}|} \sum\nolimits_{i,j} \sum\nolimits_{k=1}^{K} Ric(i,j) \pi_{ik} \pi_{jk}. \tag{3}$$

Similarly, the inter-cluster density is given as

$$D_{inter} = \frac{1}{|\mathcal{E}|K} \sum\nolimits_{i,j} \sum\nolimits_{k_1 \neq k_2} Ric(i,j) \pi_{ik_1} \pi_{jk_2}. \tag{4}$$

Consequently, the Ricci loss is defined as follows,

$$\mathcal{L}_{Ric} = \alpha_0 D_{inter} - D_{intra}, \tag{5}$$

where α_0 is a weighting coefficient. The rationale of our formulation is that we maximize node density within the cluster while minimizing the density across different clusters.

Connection to the Famous Ricci Flow. In differential geometry, the Ricci flow approach is to divide a smooth manifold into different regions based on the Ricci curvature. The regions of large positive curvature shrink in whereas regions of very negative curvature spread out [Chen and Zhu, 2005]. Analogy to the smooth manifold, we divide a graph into different node clusters where *positive Ricci curvature groups the nodes and negative Ricci departs them.*

Ni *et al.* [2019]; Sia *et al.* [2019] leverage Ricci curvatures to group nodes, but they do not consider the end-to-end clustering in a curvature space, essentially different from our setting. We are the first to introduce the curvature space to the problem of graph clustering to the best of out knowledge.

3.2 Constructing Heterogeneous Curvature Space

We are facing a challenging task: constructing a new curvature space for the geometric graph clustering. Most existing graph curvature spaces *present as a single curvature radius* (either the typical hyperbolic, spherical and Euclidean spaces or the recent ultrahyperbolic and quotient manifolds [Xiong *et al.*, 2022b; Law, 2021]). However, rather than a single curvature, geometric clustering requires a closer look over the various fine-grained curvatures on the graph.

A core contribution of our work is that we introduce a novel *heterogeneous curvature space*, bridging this gap. In a nutshell, it is a product space of learnable factor manifolds and multiple free coordinates, as shown in Fig 1 (b).

A Novel Product Manifold

We introduce the intuition of our idea before the formal theory. The graph curvature spaces above are restricted by a fixed norm, thus yielding a single curvature radius. We enrich the curvatures by producting a single radius space with **multiple free coordinates** that do not have any norm restriction. (A more theoretical rationale based on rotational symmetry [Giovanni *et al.*, 2022] is given in Appendix.) Our heterogeneous curvature space $\mathcal{P}_H$ is constructed as follows,

$$\mathcal{P}_H = \otimes_{m=0}^{M} \mathcal{M}_m^{c_m, d_m}, \ \mathcal{M}_0^{c_0, d_0} := \mathbb{R}^{d_0}, c_0 = 0, \tag{6}$$

where $\otimes$ denotes the Cartesian product. It is a product of M *restricted factors* and *a free factor of* d_0 *free coordinates*. In the product space, a point $\boldsymbol{z} \in \mathcal{P}_H$ is thus expressed as the concatenation of its factors $\boldsymbol{z}^m \in \mathcal{M}_m^{c_m,d_m}$ with the combinational distance metric of $d^2_{\mathcal{P}}(\boldsymbol{x},\boldsymbol{y}) = \sum_m d^2_{c_m}(\boldsymbol{x}^m,\boldsymbol{y}^m)$.

A restricted factor $\mathcal{M}_m^{c_m,d_m}$ is defined on the manifold,

$$\left\{ \boldsymbol{z} = \begin{bmatrix} z_t \\ \boldsymbol{z}_s \end{bmatrix} \middle| \ \langle \boldsymbol{z},\boldsymbol{z}\rangle_{c_m} = \frac{1}{c_m},\ z_t \in \mathbb{R}, \boldsymbol{z}_s \in \mathbb{R}^{d_m} \right\}, \quad (7)$$

with the metric inner product $\langle \boldsymbol{z},\boldsymbol{z}\rangle_{c_m} = sgn(c_m) z_t^2 + \boldsymbol{z}_s^\top \boldsymbol{z}_s$, where sgn is the sign function. c_m and d_m denote the curvature and dimension, respectively. The induced *norm restriction* is given as $\|\boldsymbol{z}\|^2_{c_m} = \langle \boldsymbol{z},\boldsymbol{z}\rangle_{c_m}$. z_t is the 1st dimension, and is usually termed as t-dimension. The north pole is $\mathbf{0} = (|c_m|^{-\frac{1}{2}}, 0, \cdots, 0)$. The closed-form distance d_{c_m}, logarithmic $log^{c_m}_{\boldsymbol{z}}$ and exponential maps $exp^{c_m}_{\boldsymbol{z}}$ are derived in Skopek *et al.* [2020]. *The free factor* $\mathbb{R}^{d_0}$ looks Euclidean like, but in fact we inject the rotational symmetry in it. The closed-form distance d_0 is given in [Giovanni *et al.*, 2022]. We do not use its logarithmic/exponential maps in our model.

We prove that the proposed $\mathcal{P}_H$ has heterogeneous curvatures, i.e., it allows for different curvatures on the different regions. Supporting curvature heterogeneity is the foundation of geometric clustering. We start with the concept below.

Definition (Diffeomorphism [Lee, 2013]). *Given two manifolds* $\mathcal{M}_1$ *and* $\mathcal{M}_2$*, a smooth map* $\varphi : \mathcal{M}_1 \to \mathcal{M}_2$ *is referred to as a diffeomorphism if* φ *is bijective and its inverse* φ^{-1} *is also smooth.* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *are said to be diffeomorphic and denoted as* $\mathcal{M}_1 \simeq \mathcal{M}_2$ *if there exists a* φ *connecting them.*

Proposition 1 (Curvature Heterogeneity). $\forall d_0 > 1, \forall c_m$*, there exists a* diffeomorphism *of* $\mathcal{P}_H \simeq (\otimes_{m=1}^{M} \mathcal{M}_m^{c_m,d_m} \otimes \mathcal{M}^{0,d}) \otimes \mathbb{R}_S$ *where a point* $\boldsymbol{z}_i$*'s curvature is a map* $\psi((\boldsymbol{z}_i)_{[S]}, c_1, \cdots, c_M)$ *w.r.t. its location with the differential operator*

$$\frac{-2\partial^2_{SS}\rho}{\rho} + \frac{1-(\partial^2_S\rho)^2}{\rho^2}, \quad (8)$$

for some smooth ρ *and* $(\boldsymbol{z}_i)_{[S]}$ *is the coordinate of* $\mathbb{R}_S$*, where* $\mathcal{M}^{0,d} \otimes \mathbb{R}_S = \mathbb{R}^{d_0}$ *and* $\mathbb{R}_S$ *is the axis for rotational symmetry.*

Proof. Please refer to the Appendix. □

Fine-grained Curvature Modeling for Graph Clustering

Here, we derive the fine-grained node-level curvature in our product space. With the definition of *Diffeomorphism* above and *Proposition 1*, the curvature c_i of $\boldsymbol{z}_i \in \mathcal{P}_H$ can be derived from the map $(\varphi \circ \psi)((\boldsymbol{z}_i)_{[S]}, c_1, \cdots, c_M)$ and the differential operator on ρ. That is, $\boldsymbol{z}_i$'s curvature is inferred via a map regarding the curvatures of factor manifold $c_1, \cdots, c_M$ and its coordinate of rotation symmetry $(\boldsymbol{z}_i)_{[S]}$. In our construction, $(\boldsymbol{z}_i)_{[S]}$ is given in the 1st dimension of the $\boldsymbol{z}_i$'s free factor $(\boldsymbol{z}_i^0)_{[1]}$. We employ a multilayer perceptron (MLP) to approximate the map. The estimated curvature $\bar{c}_i$ is given as,

$$\bar{c}_i = MLP([(\boldsymbol{z}_i^0)_{[1]}, c_1, \cdots, c_M]^\top). \quad (9)$$

In the graph domain, node curvature $Ric(i)$ is defined by averaging the Ricci curvature in its neighborhood, in analogy to tracing around the tangent space of the manifold. That is, the node-level curvature on the graph is formulated as $Ric(i) = \frac{1}{degree_i} \sum_{j \in \mathcal{N}_i} Ric(i,j)$, where $degree_i$ is the degree of v_i and $\mathcal{N}_i$ denotes the 1-hop neighborhood of node i. Then, we propose a node-level curvature consistency loss as

$$\mathcal{L}_{Curv} = \frac{1}{N} \sum\nolimits_i |Ric(i) - \bar{c}_i|^2, \quad (10)$$

so that *curvatures of factor manifolds are jointly learnt with the model via the fine-grained curvature modeling.*

Till now, we construct the heterogeneous curvature space modeling the fine-grained curvatures of the graph. Thereby, *the constructed curvature space supports geometric graph clustering with the Ricci loss, which requires a closer look over the various Ricci curvatures on the graph* (Eqs. 3-5).

Remarks. The advantages of our design are 1) $\mathcal{P}_H$ supports node-level curvature modeling for geometric clustering, and its factors has learnable curvatures, different from the product manifolds in Gu *et al.* [2019]; Wang *et al.* [2021]. 2) $\mathcal{P}_H$ as a whole owns the closed form expression of geometric metrics inherited from its factor manifolds. 3) $\mathcal{P}_H$ decomposes itself into $(M+1)$ *different geometric views* corresponding to each factor (i.e., M *restricted views* and 1 *free view*).

Generate Deep Representations in the Product Manifold

Thanks to the product construction, encoding in the heterogeneous curvature space is transformed into encoding in each factor manifold. Most of the Riemannian GCNs involve the tangent space out of the original manifold, and recent studies observe the inferior of tangential methods [Dai *et al.*, 2021].

To bridge this gap, we design a *fully Riemannian* GCN (***fRGCN***) for the restricted factor $\mathcal{M}^{c,d}$, whose novelty lie in that *all the operations are fully Riemannian for any* c, i.e., no tangent space is involved. We design the manifold preserving operators of ***fRGCN*** as follows.

Feature Transformation. First, we formulate a generalized Lorentz Transformation (gLT) for dimension transformation, inspired by the classic LT. The transform $\mathcal{M}^{c,d_m} \to \mathcal{M}^{c,d_n}$ is done via the matrix left-multiplication with the transform matrix derived as follows,

$$gLT^{c,d_m \to d_n}_{\boldsymbol{z}}(\boldsymbol{W}) = \begin{bmatrix} w_t & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{W} \end{bmatrix}. \quad (11)$$

Recall that $\boldsymbol{z} = [z_t\ \boldsymbol{z}_s]^\top \in \mathcal{M}^{c,d_m}$. In gLT, w_t is responsible to scale z_t while $\boldsymbol{W}$ transforms $\boldsymbol{z}_s$. We derive the closed-form t-scaling as $w_t = \frac{1}{z_t}\sqrt{sgn(c)\left(\frac{1}{c} - \ell(\boldsymbol{W},\boldsymbol{z}_s)\right)}$ and $\ell(\boldsymbol{W},\boldsymbol{z}_s) = \|\boldsymbol{W}\boldsymbol{z}_s\|^2$.

Now, we prove that the transformed feature with gLT resides in the target manifold.

Proposition 2 (Manifold Preserving). $\forall \boldsymbol{z} \in \mathcal{M}^{c,d_m}, \forall c$*,* $gLT^{c,d_m \to d_n}_{\boldsymbol{z}}(\boldsymbol{W})\boldsymbol{z} \in \mathcal{M}^{c,d_n}$ *holds for any* $\boldsymbol{W} \in \mathbb{R}^{d_n \times d_m}$.

Proof. Refer to our concurrent work [Sun *et al.*, 2023]. □

Note that, the classic LT works with a fixed dimension. Recently, Dai *et al.* [2021] optimize with orthogonal constraint unfriendly to deep learning. Chen *et al.* [2022] restrict in negative curvature. That is, all of them cannot satisfy our need.

Second, we add the bias for gLT and obtain the linear layer in the manifold of any curvature c as follows,

$$LL^c(\boldsymbol{W}, \boldsymbol{z}, \boldsymbol{b}) = \begin{bmatrix} w_t z_t \\ \boldsymbol{W}\boldsymbol{z}_s + \boldsymbol{b} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{b}$ is the bias and $\ell(\boldsymbol{W}, \boldsymbol{z}_s) = \|\boldsymbol{W}\boldsymbol{z}_s + \boldsymbol{b}\|^2$. It is easy to check that LL^c is *manifold preserving*.

Attentive Aggregation. The encoding of i is updated as the weighted geometric centorid over the set $\bar{\mathcal{N}}_i$, the neighbors of i and itself, i.e., $\arg\min_{\boldsymbol{h}_i \in \mathcal{M}} \sum_{j \in \bar{\mathcal{N}}_i} \nu_{ij} d_c^2(\boldsymbol{h}_i, \boldsymbol{h}_j), \forall c$ and ν_{ij} denotes the attentive weight. For any c, we derived the closed form solution $\boldsymbol{h}_i = AGG^c(\{\boldsymbol{h}_j, \nu_{ij}\} | j \in \bar{\mathcal{N}}_i)$,

$$AGG^c(\{\boldsymbol{h}_j, \nu_{ij}\} | j \in \bar{\mathcal{N}}_i) = \frac{1}{\sqrt{|c|}} \sum_{j \in \bar{\mathcal{N}}_i} \frac{\nu_{ij} \boldsymbol{h}_j}{\left| \| \sum_{j \in \bar{\mathcal{N}}_i} \nu_{ij} \boldsymbol{h}_j \|_c \right|}. \tag{13}$$

The attentive weights ν_{ij} is the importance of j in the aggregation over $\bar{\mathcal{N}}_i$. We define the attentive weights based on the distance in the manifold, $\nu_{ij} = Softmax(-\tau d^c(\boldsymbol{h}_j, \boldsymbol{h}_i) - \gamma)$, where τ is an inverse temperature and we add a bias γ. It is easy to check that the centroid in Eq. (13) lives in the manifold, $\forall c$, and thus AGG^c is *manifold preserving*.

Note that, Einstein midpoint formulates an arithmetic mean in the manifold but lacks geometric interpretation. Fréchet mean elegantly generalizes from Einstein midpoint but does not offer any closed form solution [Chen *et al.*, 2022]. Our closed form solution in Eq. (13), generalizing to any curvature, is the geometric centroid of squared distance.

The Free Factor. Linear layer LL^0 is done via replacing LL^c with a free $w_t \in \mathbb{R}$. Attentive aggregation is defined as $AGG^0(\{\boldsymbol{h}_j, \nu_{ij}\} | j \in \bar{\mathcal{N}}_i) = \sum_{j \in \bar{\mathcal{N}}_i} \nu_{ij} \boldsymbol{h}_j$ where attentive weights ν_{ij} is computed based on distance d_0. They are manifold preserving as there is no norm restriction in $\mathbb{R}^{d_0}$.

3.3 Learning by Reweighted Geometric Contrasting

In this subsection, we train the graph clusters with a contrastive loss in the proposed curvature space. Specifically, we propose a Reweighted Geometric Contrasting (RGC) approach, in which we contrast across different geometric views with a novel dual reweighting, as shown in Fig 1 (c).

Augmentation-Free Geometric Contrast

The augmentation is nontrivial for graph contrastive learning, and requires special design for clustering [Gong *et al.*, 2022]. Instead, our CONGREGATE is free of augmentation where we take advantage of the carefully designed $\mathcal{P}_H$ for contrastive learning. Thanks to the product construction, $\mathcal{P}_H$ itself owns different *geometric views* as remarked in Sec. 3.2. The contrast strategy is that we contrast each restricted view in $\mathcal{M}_m^{c_m, d_m}$ with the free view in $\mathbb{R}^{d_0}$, and vice versa.

The remaining challenge is how to contrast between different manifolds, i.e., $\mathcal{M}_m^{c_m, d_m}$ and $\mathbb{R}^{d_0}$. The difference in both curvature and dimension blocks the application of typical similarity functions. We propose to bridge this gap by gLT and bijection $\psi_{\mathcal{M} \to \mathbb{R}}$ of *Diffeomorphism*. (Recall that we have already provided an effective mathematics tool for dimension transformation, gLT.) Specifically, we introduce an image of restricted view $\hat{\boldsymbol{z}}^m$ that is comparable with the free view. First, we employ gLT to transform $\boldsymbol{z}^m$ into $\mathcal{M}_m^{c_m, d_0 - 1}$ whose ambient space is $\mathbb{R}^{d_0}$. Second, we apply the diffeomorphism bijection and thus the image is given as follows,

$$\hat{\boldsymbol{z}}^m = \psi_{\mathcal{M} \to \mathbb{R}}(gLT_{\boldsymbol{z}^m}^{c_m, d_m \to (d_0 - 1)}(\mathbf{W}) \boldsymbol{z}^m), \tag{14}$$

where parameter $\mathbf{W}$ characterizes gLT, and $log_{\mathbf{0}}^{c_m}(\cdot)$ is utilized as the bijection since its differentiable inverse exists $log_{\mathbf{0}}^{c_m}(exp_{\mathbf{0}}^{c_m}(\boldsymbol{z})) = \boldsymbol{z}$. Note that $\hat{\boldsymbol{z}}_i^m \in \mathbb{R}^{d_0}$. Then, we define the similarity as a bilinear critic with parameter $\mathbf{S}$,

$$Sim(\boldsymbol{z}^m, \boldsymbol{z}^0) = (\hat{\boldsymbol{z}}^m)^\top \mathbf{S} \boldsymbol{z}^0. \tag{15}$$

Our formulation of Eq. (15) does not introduce additional tangent space, and its advantage is examined in Sec. 4.3.

Algorithm 1: Training CONGREGATE

Input: Graph G, #(Clusters)=K, #(Factors)=$(M+1)$
Output: Encoder f, Cluster centroids $\{\boldsymbol{\phi}_k\}_{k=1,\cdots,K}$

```
Preprocessing: Compute Ricci curvatures on G;
while not converging do
    Create geometric views [z^0 z^1 ··· z^M] ← fRGCN;
    for each restricted view z^m, m ∈ [1, M] do
        /* Contrast with the free view z^0 */
        Node-to-Node contrast based on Eq. (17);
        Node-to-Cluster contrast based on Eq. (18);
    end
    Train {φ_k}_{k=1,···,K} by optimizing J in Eq. (20);
end
```

Dual Reweighting in Curvature Space

A drawback of the popular InfoNCE loss is hardness unawareness (equally treating the hard sample pairs and the easy ones), limiting the discriminative ability [Robinson *et al.*, 2021]. To address this issue, we propose a *dual reweighting*, paying more attention to both hard negatives and hard positives for contrastive learning in curvature space.

First, we specify the hard samples in the context of graph clustering where cluster assignment offers pseudo labels. Intuitively, the nodes assigned to different clusters but sharing large similarity are referred to as *hard negatives*, while *the border nodes sharing small similarity to the cluster centroid are hard positives*. Second, we model the hardness by comparing cluster assignment (pseudo label) and representation similarity, and formulate the dual reweighting as follows,

$$\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{z}_j^0) = |\boldsymbol{\pi}_i{}^\top \boldsymbol{\pi}_j - Sim(\hat{\boldsymbol{z}}_i^m, \boldsymbol{z}_j^0)|^\beta \tag{16}$$

where the control coefficient β is a positive integer, and $\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{z}_j^0)$ up-weights both hard positives and hard negatives while down-weighting the easy ones.

Recently, Sun *et al.* [2022a] design a Riemannian reweighing for node embedding only and thus fail to consider clusters. Liu *et al.* [2023] select hard positives in Euclidean space while we need to handle different manifolds. Both of them cannot meet our need and motivate our design of Eq. (16).

Node-to-Node & Node-to-Cluster Contrasting

The RGC loss consists of Node-to-Node and Node-to-Cluster contrasting, where we contrast different geometric views with the dual reweighting and Sim function in generic curvature space. First, we define Node-to-Node contrast loss as follows,

$$I(\mathbf{Z}^m, \mathbf{Z}^0) = -\sum_{i=1}^{N} log \frac{e^{\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{z}_i^0) Sim(\boldsymbol{z}_i^m, \boldsymbol{z}_i^0)}}{\sum_{j=1}^{N} e^{\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{z}_j^0) Sim(\boldsymbol{z}_i^m, \boldsymbol{z}_j^0)}}. \tag{17}$$

Second, we contrast node encoding of one view with cluster centroids of another view, and formulate the Node-to-Cluster contrast loss as follows,

$$I(\mathbf{Z}^m, \mathbf{\Phi}^0) = -\sum_{i=1}^{N} log \frac{e^{\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{\phi}_{k_i}^0) Sim(\boldsymbol{z}_i^m, \boldsymbol{\phi}_{k_i}^0)}}{\sum_{k=1}^{K} e^{\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{\phi}_k^0) Sim(\boldsymbol{z}_i^m, \boldsymbol{\phi}_k^0)}}, \tag{18}$$

where node v_i is assigned to cluster k_i. Here, in $\mathcal{W}(\boldsymbol{z}_i^m, \boldsymbol{\phi}_{k_i}^0)$, the inner product term is simplified as $[\boldsymbol{\pi}_i]_{k_i}$ the probability of v_i assigned to cluster k_i. Thus, we have RGC loss as follows,

$$\mathcal{L}_{RGC} = \sum_{m=1}^{M} \sum_{\mathbf{X} \in \{\mathbf{Z}^0, \mathbf{\Phi}^0\}} (I(\mathbf{Z}^m, \mathbf{X}) + I(\mathbf{X}, \mathbf{Z}^m)). \tag{19}$$

In our curvature space, intra-cluster node similarity is maximized as they positively contrast to the same centroid, while inter-cluster nodes are separated by negative contrast. Meanwhile, more attention is paid to the similar cluster centorids (*hard negatives*) and the nodes residing in the cluster border (*hard positives*), thanks to dual reweighting of Eq. (16).

Overall Loss. Finally, the overall loss of CONGREGATE is formulated as follows,

$$\mathcal{J} = \mathcal{L}_{Ric} + \alpha_1 \mathcal{L}_{Curv} + \alpha_2 \mathcal{L}_{RGC}, \tag{20}$$

where α_1 and α_2 are weighting coefficients. In this way, we end-to-end train node encodings and cluster centorids, so that the nodes in the graph are clustered in the proposed heterogeneous curvature space.

Complexity Analysis

We summarize the training process of CONGREGATE in Algorithm 1, in which Eq. (19) is the most costly, yielding the computational complexity of $O(2M|\mathcal{V}|^2 + 2MK|\mathcal{V}|)$. Note that, the computational complexity is similar to typical contrastive methods [Veličković *et al.*, 2019; Hassani and Ahmadi, 2020]. The Ricci curvatures only need to be computed once as a pre-processing, and can be effectively obtained similar to Ni *et al.* [2019]; Ye *et al.* [2020].

4 Experiment

In this section, we evaluate our model with 19 baselines on 4 public datasets, aiming to answer the following research questions (*RQs*),

- ***RQ1***: How does the proposed CONGREGATE perform?
- ***RQ2***: What are the effects of the proposed components?
- ***RQ3***: Why does *Ricci Curvature* work?

Datasets	Edge Type	# Nodes	# Edges	# Classes
Cora	citation	2,708	5,429	7
Citeseer	citation	3,327	9,104	6
MAG-CS	co-author	18,333	163,788	15
AMAP	co-purchase	7,487	119,043	8

Table 1: The statistics of the datasets

4.1 Experimental Setups

Datasets & Baselines. To evaluate the proposed model, we choose 4 public datasets, i.e., Cora and Citeseer [Devvrit *et al.*, 2022], and larger MAG-CS [Park *et al.*, 2022] and Amazon-Photo [Li *et al.*, 2022]. We give the statistics of the datasets in Table 1.

We focus on deep graph clustering with no labels available. Thus, both the strong deep clustering methods (*DC*) and self-supervised learning methods (*SS*) are included as *Euclidean Baselines* for a comprehensive evaluation. There are 13 strong DC methods and 5 SS methods, summarized in Table 2. Specifically, SS methods are GAE, VGAE [Kipf and Welling, 2016], ARGA [Pan *et al.*, 2020], and the recent contrastive ones, DGI [Veličković *et al.*, 2019] and MVGRL [Hassani and Ahmadi, 2020]. DC methods are DAEGC [Wang *et al.*, 2019], SDCN [Bo *et al.*, 2020], AGE [Cui *et al.*, 2020], GMM-VGAE [Hui *et al.*, 2020], AGCN [Peng *et al.*, 2021], GDCL [Zhao *et al.*, 2021], S^3GC [Devvrit *et al.*, 2022], CGC [Park *et al.*, 2022], gCooL [Li *et al.*, 2022], HostPool [Duval and Malliaros, 2022], AGC-DRR [Gong *et al.*, 2022], FT-VGAE [Mrabah *et al.*, 2022] and HSAN [Liu *et al.*, 2023]. There exists few *Riemannian Baselines* (*R*). Note that, recent Riemannian GNNs do not have clustering ability, as typical clustering algorithms cannot be directly applied/incorporated owing to the inherent difference in geometry. Instead, we choose a recent shallow model, RicciCom [Ni *et al.*, 2019]. We are the first to bridge Riemannian space and graph clustering to our knowledge.

Evaluation Metric. We employ 3 popular evaluation metrics, i.e., Normalized Mutual Information (NMI), Adjusted Rand Index (ARI) and Accuracy (ACC) [Devvrit *et al.*, 2022; Li *et al.*, 2022; Mrabah *et al.*, 2022].

Reproducibility. In our model, the number of restricted factors M and their dimensions need to be configured for an instantiation, while the factors' curvatures are jointly learnt with the model. In ***fRGCN***, the convolutional layer is stacked twice. The parameters living in the factor (e.g., $\mathbf{W}$ in gLT, and the centroid in the factors) are optimized via Riemannian Adam [Kochurov *et al.*, 2020]. The grid search is performed over search spaces for the hyperparameters, e.g., learning rate: $[0.001, 0.003, 0.005, 0.008, 0.01]$, dropout rate: $[0.0, 0.1, 0.2, 0.3, 0.4]$. We utilize a 2-layer MLP to approximate the fine-grained curvature. In RGC loss, hyperparameter β of the reweighting is 2 as default. If input features live in the Euclidean space, we use the inverse bijection $\psi^{-1}_{\mathcal{M}\to\mathbb{R}}$ in Eq. (14) to map the Euclidean input to a factor manifold. Further details and code are provided https://github.com/CurvCluster/Congregate.

Method	Cora			Citeseer			MAG-CS			Amazon-Photo		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
GAE	61.3 (0.8)	44.4 (1.1)	38.1 (0.9)	61.4 (0.8)	34.6 (0.7)	33.6 (1.2)	63.2 (2.6)	69.9 (0.6)	52.8 (1.5)	71.6 (2.5)	62.1 (2.8)	48.8 (4.6)
VGAE	64.7 (1.3)	43.4 (1.6)	37.5 (2.1)	61.0 (0.4)	32.7 (0.3)	33.1 (0.5)	60.4 (2.9)	65.3 (1.4)	50.0 (2.1)	74.3 (3.6)	66.0 (3.4)	56.2 (4.7)
DGI	72.6 (0.9)	57.1 (1.7)	51.1 (3.0)	68.6 (1.3)	43.5 (1.2)	44.5 (1.9)	60.0 (0.6)	65.9 (0.4)	50.3 (0.9)	57.2 (1.9)	37.6 (0.3)	26.4 (0.3)
ARGA	71.0 (2.5)	51.1 (0.5)	47.7 (0.3)	61.1 (0.5)	34.4 (0.7)	33.4 (1.2)	47.9 (6.0)	48.7 (3.0)	23.6 (9.0)	69.3 (2.3)	58.4 (2.8)	44.2 (4.4)
MVGRL	70.5 (3.7)	55.6 (1.5)	48.7 (3.9)	62.8 (1.6)	40.7 (0.9)	34.2 (1.7)	61.6 (3.3)	65.4 (1.9)	49.2 (5.3)	41.1 (3.2)	30.3 (3.9)	18.8 (2.3)
DAEGC	70.4 (0.4)	52.9 (0.7)	49.6 (0.4)	64.5 (1.4)	36.4 (0.9)	37.8 (1.2)	48.1 (3.8)	60.3 (0.8)	47.4 (4.2)	76.0 (0.2)	65.3 (0.5)	58.1 (0.2)
SDCN	35.6 (2.8)	14.3 (1.9)	7.8 (3.2)	65.9 (0.3)	38.7 (0.3)	40.2 (0.4)	51.6 (5.5)	58.0 (1.9)	46.9 (8.1)	53.4 (0.8)	44.9 (0.8)	31.2 (1.2)
AGE	73.5 (1.8)	57.6 (1.4)	50.1 (2.1)	69.7 (0.2)	44.9 (0.5)	45.3 (0.4)	59.1 (1.7)	66.7 (0.3)	51.1 (2.8)	75.9 (0.7)	65.4 (0.6)	55.9 (1.3)
GMM	71.5 (0.2)	53.1 (2.1)	47.4 (0.6)	67.5 (0.9)	40.7 (1.1)	42.4 (0.5)	67.2 (2.6)	72.8 (0.7)	56.1 (1.9)	75.5 (0.3)	68.1 (0.7)	57.9 (0.9)
AGCN	72.2 (3.6)	54.7 (1.3)	48.9 (2.7)	68.8 (0.2)	41.5 (0.3)	43.8 (0.3)	54.2 (5.2)	59.4 (2.1)	49.2 (6.5)	45.2 (1.0)	41.6 (1.1)	36.6 (0.2)
GDCL	70.8 (0.5)	56.6 (0.4)	48.1 (0.7)	66.4 (0.6)	39.5 (0.4)	41.1 (1.0)	53.9 (3.1)	60.3 (0.8)	48.8 (5.1)	43.8 (0.8)	37.3 (0.3)	21.6 (0.5)
S^3GC	74.2 (3.0)	58.9 (1.8)	54.4 (2.5)	68.8 (1.7)	44.1 (0.9)	44.8 (0.6)	65.4 (2.3)	77.6 (0.6)	61.9 (2.4)	71.8 (0.2)	63.7 (0.8)	45.8 (1.0)
CGC	73.1 (2.2)	57.0 (0.9)	49.3 (1.8)	69.6 (0.6)	44.6 (0.1)	46.0 (0.6)	69.3 (4.0)	79.3 (1.2)	64.4 (3.7)	75.2 (0.1)	64.1 (1.2)	51.7 (0.6)
gCooL	72.0 (1.6)	58.3 (0.2)	56.9 (0.9)	71.5 (1.0)	47.3 (0.8)	46.8 (1.5)	70.7 (1.5)	78.6 (1.0)	66.0 (1.6)	72.7 (0.6)	63.2 (0.0)	52.4 (0.2)
HostPool	71.8 (0.8)	60.5 (1.0)	58.5 (1.1)	70.9 (0.5)	50.2 (0.3)	45.9 (0.7)	67.5 (2.1)	79.0 (0.6)	67.1 (1.2)	69.4 (0.1)	59.8 (1.0)	45.5 (0.7)
AGC-DRR	40.6 (0.6)	18.7 (0.7)	14.8 (1.6)	68.3 (1.8)	43.3 (1.4)	45.3 (2.3)	71.2 (0.8)	71.2 (0.8)	65.8 (3.1)	76.8 (1.5)	66.5 (1.2)	60.1 (1.6)
FT-VGAE	77.4 (1.1)	61.0 (0.5)	58.2 (1.3)	70.8 (0.5)	44.5 (0.1)	46.7 (0.7)	73.3 (1.0)	69.5 (0.5)	67.9 (2.2)	78.1 (1.0)	69.8 (0.7)	59.3 (0.8)
HSAN	77.1 (1.6)	59.2 (1.0)	57.5 (2.7)	71.2 (0.8)	45.1 (0.7)	47.1 (1.1)	72.8 (1.0)	77.3 (0.9)	68.2 (1.7)	77.0 (0.3)	67.2 (0.3)	58.0 (0.5)
RiccCom	55.6 (0.3)	58.1 (1.1)	48.9 (0.9)	67.3 (1.2)	46.2 (0.8)	44.9 (0.6)	69.1 (3.2)	71.6 (1.2)	65.2 (0.8)	58.3 (2.3)	62.9 (0.9)	57.4 (0.9)
Congregate	**78.5** (1.0)	**63.2** (0.5)	**59.3** (1.2)	**72.7** (0.6)	**50.9** (0.3)	**48.3** (1.0)	**73.3** (0.7)	**80.8** (1.3)	**69.5** (1.1)	**79.7** (1.8)	**71.3** (0.5)	**60.9** (1.1)

Table 2: Clustering results of 20 methods on Cora, Citerseer, MAG-CS and Amazon-Photo in terms of NMI, ARI and ACC (%). Standard derivation is given in brackets. The best results are highlighted in **bold**, and the runner up underlined.

4.2 *RQ1*: Main Results

The clustering results on all the datasets in terms of NMI, ARI and ACC are reported in Table 2. We perform 10 independent runs, and report the mean value with standard deviation for fair comparisons. The number of clusters K is set as the number of real classes on each dataset. For the encoding-clustering baselines, we apply K-means to obtain the results. In Table 2, our CONGREGATE is instantiated with 4 factor manifolds whose dimensionality are $\{32, 32, 16, 16\}$, and it consistently achieves the best results among 19 competitors on 4 datasets. The reasons are two-fold. 1) We take advantage of the proposed curvature space and the consensus clustering from different geometric views. 2) We learn high discriminative encodings and cluster centroids in an end-to-end fashion.

4.3 *RQ2*: Ablation Study

We investigate on how each proposed component contributes to the success of our CONGREGATE: i) ***fRGCN*** for modeling graph fully Riemannianly, ii) $\varphi \circ gLT$ for contrasting between different manifolds and iii) the dual reweighting in $\mathcal{W}$ for paying attention to hard samples.

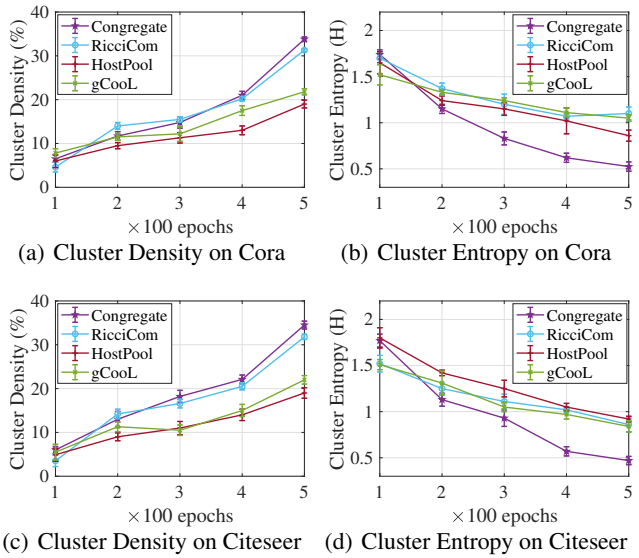
To evaluate the effectiveness of ***fRGCN***, we introduce a variant which replaces ***fRGCN*** with a GAT^c. Concretely, GAT^c generalizes GAT [Veličković *et al.*, 2018] in a manifold of curvature c with tangent spaces. We utilize the tangential methods of any c formulated in Skopek *et al.* [2020]. To evaluate the effectiveness of $\varphi \circ gLT$, we introduce a variant using $\mathbf{T} log_{\mathbf{0}}^{c_m}$ instead, where the matrix $\mathbf{T}$ is given for dimension transformation. It introduce an additional tangent space compare to the design in our model. To evaluate the effectiveness of $\mathcal{W}$, we introduce two kinds of variants. The first variant (denoted as $-pAware$) removes the $\mathcal{W}$ on the numerators of our RGC loss, thus keeping the attention to hard negatives only. The second variant (denoted as $-hard$) eliminates all the $\mathcal{W}$, resulting in an InfoNCE in Riemannian space without hardness awareness. In addition, we examine the effect on the number of factor manifolds. To this end, the variants above are instantiated in product space of 4 factors and 5 factors, respectively. We report the NMI and ACC of the clustering results on Cora and Citeseer datasets in Table 2, and find that: **i)** Our CONGREGATE beats the $-$***fRGCN*** and $-\varphi \circ gLT$. It shows that introducing addtional tangent spaces trends to results in inferior clustering, and thus *testifies the effectiveness of fully Riemannian model.* **ii)** The product space 5 factors outperforms that of 4 factors. Here, the number of factors corresponds to the number of curvatures. It suggests that more factor manifolds may benefit the performance, and the reason is that more factors give further flexibility for the fine-grained curvature modeling. **iii)** $-posA$ variant performs better than $-hard$, and the proposed RGC loss is the best. It shows the importance of hard samples, and *more attentions to hard positives (the border nodes) further help the performance*, which is the reason of our design that we pay more attentions to both hard positives and hard negatives.

	Variant	Cora		Citeseer	
		ACC	NMI	ACC	NMI
4 factors	CONGREGATE	**78.5** (1.0)	**63.2** (0.5)	**72.7** (0.6)	**50.9** (0.3)
	$-f$RGCN	75.9 (0.5)	60.4 (0.2)	72.0 (0.9)	48.3 (0.6)
	$-\varphi \circ gLT$	77.5 (0.8)	61.7 (0.6)	71.9 (1.3)	47.9 (0.2)
	$-pAware$	77.2 (0.7)	62.1 (0.8)	70.3 (0.5)	49.0 (1.4)
	$-hard$	76.8 (1.1)	61.5 (0.3)	69.8 (0.2)	48.9 (0.5)
5 factors	CONGREGATE	**78.1** (0.9)	**63.8** (0.4)	**73.1** (0.6)	**52.4** (0.7)
	$-f$RGCN	76.3 (1.2)	61.9 (0.5)	72.3 (1.0)	49.8 (0.9)
	$-\varphi \circ gLT$	77.8 (0.6)	62.5 (0.9)	72.8 (0.7)	51.6 (0.4)
	$-pAware$	77.3 (2.2)	63.0 (0.3)	71.2 (1.1)	51.2 (1.0)
	$-hard$	76.5 (1.3)	62.2 (0.7)	70.6 (0.5)	49.5 (0.8)

Table 3: Ablation study on Cora and Citeseer datasets.

4.4 *RQ3*: Ricci Curvature & Clustering

We discuss why Ricci curvature works. Empirically, we further study the clustering capability of Ricci curvature comparing with classic concepts ($gCooL$ with refined modularity, *HostPool* with motif conductance and *RicciCom* with Ricci curvature). We examine the result clusters from mi-



(a) Cluster Density on Cora (b) Cluster Entropy on Cora

(c) Cluster Density on Citeseer (d) Cluster Entropy on Citeseer

Figure 2: Visualize density and entropy of the clusters.

croscopic perspective by cluster density and entropy [Li *et al.*, 2022]. The density is $\mathbb{E}_k[\frac{E_k}{V_k(V_k-1)}]$, where E_k and V_k are the number of edges and nodes in cluster k. The entropy is $-\mathbb{E}_k[\sum_c p_k(c)\log p_k(c)]$, where $p_k(c)$ is the frequency of class (label) c occurred in cluster k. Lower entropy means better result, i.e., the cluster contains a major class. The results are visualized in Fig. 2. After a few hundred epochs, Ricci methods achieves even better density than modularity/conductance methods. *It shows the clustering capability of Ricci curvature, verifying our motivation.* Also, we have lower entropy than RicciCom. It is because we further introduce the novel curvature space, supporting fine-grained curvature modeling for graph clustering.

5 Related Work

Deep Graph Clustering. In the literature, deep graph clustering methods are roughly divided into 3 categories regarding the learning paradigm. 1) Reconstructive methods provide supervision signal by recovering graph information, and generate node clusters by applying or incorporating clustering methods [Duval and Malliaros, 2022; Mrabah *et al.*, 2022]. 2) Adversarial methods regulate the generative process by a discriminator in a min-max game [Jia *et al.*, 2019; Yang *et al.*, 2020]. 3) Contrastive methods acquire discriminative representation without labels by pulling positive samples together while pushing negative samples apart [Park *et al.*, 2022; Devvrit *et al.*, 2022]. Meanwhile, deep methods are introduced to bipartite graphs [Zhou *et al.*, 2022], signed graphs [He *et al.*, 2022; Kang *et al.*, 2021], temporal graphs [Yao and Joe-Wong, 2021], heterogeneous graphs [Khan and Kleinsteuber, 2022], and etc. Recently, He *et al.* [2021] present a novel generative model with EM algorithm; Fettal *et al.* [2022] introduce a strong matrix optimization framework. Distinguishing from the prior studies, we rethink the problem of graph clustering from the geometric perspective.

Riemannian Graph Representation Learning. Recent years have witnessed the remarkable success achieved by Riemannian graph learning. As hyperbolic space is well aligned with hierarchical or power-law graphs, shallow models are first introduced [Nickel and Kiela, 2017; Suzuki *et al.*, 2019], and hyperbolic GCNs with different formulations are then proposed [Chami *et al.*, 2019; Liu *et al.*, 2019; Zhang *et al.*, 2021]. Beyond hyperbolic space, κ-GCN [Bachmann *et al.*, 2020] extend GCN to constant-curvature spaces with κ-sterographical model. Yang *et al.* [2022] model the graph in the dual space of Euclidean and hyperbolic ones. Xiong *et al.* [2022a,b] study graph learning on a kind of pseudo Riemannian manifold, ultrahyperbolic space. Law [2021] introduce a quotient manifold for graph learning. Both ultrahyperbolic and quotient manifolds present a single curvature radius. Cruceru *et al.* [2021] study the matrix manifold of Riemannian spaces. Gu *et al.* [2019]; Wang *et al.* [2021]; Sun *et al.* [2022b] explore node embedding in the product manifold. Our work is based on the product manifold, but we further explore the curvature heterogeneity. Sun *et al.* [2022a]; Yang *et al.* [2021]; Sun *et al.* [2021] consider representation learning on temporal graphs. Very recently, Giovanni *et al.* [2022] investigate in the rotational symmetry of the manifold, but do not consider fine-grained curvature modeling and learnable factors, different from our study. However, none of existing studies focus on graph clustering in Riemannian manifolds to the best of our knowledge.

6 Conclusion

In this paper, we formulate the problem of geometric graph clustering, which is *the first to introduce the curvature space allowing for fine-grained curvature modeling to graph clustering*. We present an end-to-end CONGREGATE built upon a novel heterogeneous curvature space that we construct for geometric graph clustering with Ricci curvatures. Accordingly, graph clusters are trained by an augmentation-free contrastive loss, where we pay more attention to both hard positives and hard negatives in our curvature space. The empirical results show the superiority of our model.

Acknowledgments

Thanks to the anonymous reviewers. The authors were supported in part by National Natural Science Foundation of China under Grant 62202164, the National Key R&D Program of China through grant 2021YFB1714800, S&T Program of Hebei through grant 21340301D and the Fundamental Research Funds for the Central Universities 2022MS018. Prof. Philip S. Yu is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941. Corresponding Authors: Li Sun and Hao Peng.

References

Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *Proceedings of ICML*, volume 119, pages 486–496, 2020.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of WWW*, pages 1400–1410. ACM / IW3C2, 2020.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in NeurIPS*, pages 4869–4880, 2019.

Bing-Long Chen and Xi-Ping Zhu. Ricci flow with surgery on four-manifolds with positive isotropic curvature. *Journal of Differential Geometry*, pages 177–264, 2005.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th ACL*, pages 5672–5686. ACL, 2022.

Calin Cruceru, Gary Bécigneul, and Octavian-Eugen Ganea. Computationally tractable riemannian manifolds for graph embeddings. In *Proceedings of AAAI*, pages 7133–7141. AAAI Press, 2021.

Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD*, pages 976–985. ACM, 2020.

Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *Proceedings of CVPR*, pages 154–163, 2021.

Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. S^3GC: Scalable self-supervised graph clustering. In *Advances in 36th NeurIPS*, 2022.

Alexandre Duval and Fragkiskos D. Malliaros. Higher-order clustering and pooling for graph neural networks. In *Proceedings of the 31st CIKM*, pages 426–435. ACM, 2022.

Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. Efficient graph convolution for joint node representation learning and clustering. In *Proceedings of the 15th WSDM*, pages 289–297. ACM, 2022.

Francesco Di Giovanni, Giulia Luise, and Michael M. Bronstein. Heterogeneous manifolds for curvature-aware graph embedding. In *Proceedings of the 10th ICLR (GTRL Workshop)*, 2022.

Lei Gong, Sihang Zhou, Wenxuan Tu, and Xinwang Liu. Attributed graph clustering with dual redundancy reduction. In *Proceedings of the 31st IJCAI*, pages 3015–3021. ijcai.org, 2022.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *Proceedings of ICLR*, pages 1–21, 2019.

Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of ICML*, volume 119, pages 4116–4126, 2020.

Dongxiao He, Shuai Li, Di Jin, Pengfei Jiao, and Yuxiao Huang. Self-guided community detection on networks with missing edges. In *Proceedings of the 30th IJCAI*, pages 3508–3514. ijcai.org, 2021.

Yixuan He, Gesine Reinert, Songchao Wang, and Mihai Cucuringu. SSSNET: semi-supervised signed network clustering. In *Proceedings of SDM*, pages 244–252. SIAM, 2022.

Binyuan Hui, Pengfei Zhu, and Qinghua Hu. Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning. In *Proceedings of the 34th AAAI*, pages 4215–4222. AAAI Press, 2020.

Yuting Jia, Qinqin Zhang, Weinan Zhang, and Xinbing Wang. Communitygan: Community detection with generative adversarial nets. In *Proceedings of the WWW*, pages 784–794. ACM, 2019.

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

Yoonsuk Kang, Woncheol Lee, Yeon-Chang Lee, Kyungsik Han, and Sang-Wook Kim. Adversarial learning of balanced triangles for accurate community detection on signed networks. In *Proceedings of ICDM*, pages 1150–1155. IEEE, 2021.

Rayyan Ahmad Khan and Martin Kleinsteuber. A framework for joint unsupervised learning of cluster-aware embedding for heterogeneous networks. In *Proceedings of the 15th WSDM*. ACM, 2022.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NeurIPS Bayesian Deep Learning Workshop*, 2016.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch. In *Proceedings of ICML (GRLB Workshop)*. PMLR, 2020.

Marc Law. Ultrahyperbolic neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 22058–22069, 2021.

John M. Lee. *Introduction to Smooth Manifolds (2nd Edition)*. Springer, 2013.

Jia Li, Jianwei Yu, Jiajin Li, Honglei Zhang, Kangfei Zhao, Yu Rong, Hong Cheng, and Junzhou Huang. Dirichlet graph variational autoencoder. In *Advances in NeurIPS*, 2020.

Bolian Li, Baoyu Jing, and Hanghang Tong. Graph communal contrastive learning. In *Proceedings of The ACM Web Conference*, pages 1203–1213. ACM, 2022.

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

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in NeurIPS*, pages 8228–8239, 2019.

Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI*, 2023.

Nairouz Mrabah, Mohamed Bouguessa, and Riadh Ksantini. Escaping feature twist: A variational graph auto-encoder for node clustering. In *Proceedings of the 31st IJCAI*, pages 3351–3357. ijcai.org, 2022.

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

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in NeurIPS*, pages 6338–6347, 2017.

Shirui Pan, Ruiqi Hu, Sai-Fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning graph embedding with adversarial training methods. *IEEE Trans. on Cybern.*, 50(6):2475–2487, 2020.

Namyong Park, Ryan A. Rossi, Eunyee Koh, Iftikhar Ahamath Burhanuddin, Sungchul Kim, Fan Du, Nesreen K. Ahmed, and Christos Faloutsos. CGC: contrastive graph clustering for community detection and tracking. In *Proceedings of the Web Conference*, pages 1115–1126, 2022.

Zhihao Peng, Hui Liu, Yuheng Jia, and Junhui Hou. Attention-driven graph clustering network. In *Proceedings of ACM MM*, pages 935–943. ACM, 2021.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *Proceedings of the 9th ICLR*. OpenReview.net, 2021.

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

Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *Proceedings of ICLR*, pages 1–54, 2020.

Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and Philip S. Yu. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of AAAI*, pages 4375–4383, 2021.

Li Sun, Junda Ye, Hao Peng, and Philip S. Yu. A self-supervised riemannian GNN with time varying curvature for temporal graph learning. In *Proceedings of the 31st CIKM*, pages 1827–1836. ACM, 2022.

Li Sun, Zhongbao Zhang, Junda Ye, Hao Peng, Jiawei Zhang, Sen Su, and Philip S. Yu. A self-supervised mixed-curvature graph neural network. In *Proceedings of AAAI*, pages 4146–4155, 2022.

Li Sun, Junda Ye, Hao Peng, Feiyang Wang, and Philip S. Yu. Self-supervised continual graph learning in adaptive riemannian spaces. In *Proceedings of AAAI*, 2023.

Ryota Suzuki, Ryusuke Takahama, and Shun Onoda. Hyperbolic disk embeddings for directed acyclic graphs. In *Proceedings of ICML*, pages 6066–6075, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of ICLR*, pages 1–12, 2018.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *Proceedings of ICLR*, pages 1–24, 2019.

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. In *Proceedings of the 28th IJCAI*, pages 3670–3676. ijcai.org, 2019.

Shen Wang, Xiaokai Wei, Cícero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, Philip S. Yu, and Isabel F. Cruz. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of The ACM Web Conference*, pages 1761–1771. ACM / IW3C2, 2021.

Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *Proceedings of ICML*, volume 162, pages 24332–24346. PMLR, 2022.

Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. Ultrahyperbolic knowledge graph embeddings. In *Proceedings of KDD*, pages 2130–2139. ACM, 2022.

Bo Xiong, Shichao Zhu, Nico Potyka, Shirui Pan, Chuan Zhu, and Steffen Staab. Pseudo-riemannian graph convolutional networks. In *Advances in 36th NeurIPS*, pages 1–21, 2022.

Liang Yang, Yuexue Wang, Junhua Gu, Chuan Wang, Xiaochun Cao, and Yuanfang Guo. JANE: jointly adversarial network embedding. In *Proceedings of the 29th IJCAI*, pages 1381–1387. ijcai.org, 2020.

Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of KDD*, pages 1975–1985. ACM, 2021.

Haoran Yang, Hongxu Chen, Shirui Pan, Lin Li, Philip S. Yu, and Guandong Xu. Dual space graph contrastive learning. In *Proceedings of The ACM Web Conference*, pages 1238–1247, 2022.

Yuhang Yao and Carlee Joe-Wong. Interpretable clustering on dynamic graphs with recurrent graph neural networks. In *Proceedings of the 35th AAAI*, pages 4608–4616. AAAI Press, 2021.

Ze Ye, Kin Sum Liu, Tengfei Ma, Jie Gao, and Chao Chen. Curvature graph network. In *Proceedings of the 8th ICLR*, 2020.

Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD*, pages 555–564. ACM, 2017.

Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proceedings of WWW*, pages 1249–1261, 2021.

Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. Graph debiased contrastive learning with joint representation clustering. In *Proceedings of the 30th IJCAI*, pages 3434–3440. ijcai.org, 2021.

Cangqi Zhou, Yuxiang Wang, Jing Zhang, Jiqiong Jiang, and Dianming Hu. End-to-end modularity-based community co-partition in bipartite networks. In *Proceedings of the 31st CIKM*, pages 2711–2720. ACM, 2022.