

Motif-Aware Riemannian Graph Neural Network with Generative-Contrastive Learning

Li Sun¹, Zhenhao Huang¹, Zixi Wang¹, Feiyang Wang², Hao Peng^{3*}, Philip Yu⁴

¹North China Electric Power University, Beijing 102206, China

²Beijing University of Posts and Telecommunications, Beijing 100876, China

³Beihang University, Beijing 100191, China

⁴Department of Computer Science, University of Illinois at Chicago, IL 60607, USA
ccsunli@ncepu.edu.cn; penghao@buaa.edu.cn; psyu@uic.edu

Abstract

Graphs are typical non-Euclidean data of complex structures. In recent years, Riemannian graph representation learning has emerged as an exciting alternative to Euclidean ones. However, Riemannian methods are still in an early stage: most of them present a single curvature (radius) regardless of structural complexity, suffer from numerical instability due to the exponential/logarithmic map, and lack the ability to capture motif regularity. In light of the issues above, we propose the problem of *Motif-aware Riemannian Graph Representation Learning*, seeking a numerically stable encoder to capture motif regularity in a diverse-curvature manifold without labels. To this end, we present a novel Motif-aware Riemannian model with Generative-Contrastive learning (MotifRGC), which conducts a minmax game in Riemannian manifold in a self-supervised manner. First, we propose a new type of Riemannian GCN (D-GCN), in which we construct a diverse-curvature manifold by a product layer with the diversified factor, and replace the exponential/logarithmic map by a stable kernel layer. Second, we introduce a motif-aware Riemannian generative-contrastive learning to capture motif regularity in the constructed manifold and learn motif-aware node representation without external labels. Empirical results show the superiority of MotifRGC.

Introduction

Graphs are the natural descriptions of real systems, ranging from social networks and recommender systems to chemistry and bioinformatics. Graph representation learning shows fundamental importance in a variety of learning tasks, such as node classification and link prediction (Hamilton, Ying, and Leskovec 2017; Velickovic et al. 2018). Euclidean space has been the workhorse of graph representation learning for decades (Perozzi, Al-Rfou, and Skiena 2014; Kipf and Welling 2017). It is not until recently that **Riemannian spaces** have emerged as an exciting alternative (Shimizu, Mukuta, and Harada 2021; Li et al. 2022a; Topping et al. 2022; Yang et al. 2023b), since they generally better match the geometry of the graph than the Euclidean counterpart (Petersen 2016; Krioukov et al. 2010). While achieving encouraging performance, Riemannian graph representation learning is still in its early stages. There are several important issues largely remaining open.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

	Curvature Diversity	Motif Regularity	Numerical Stability
HGCN (NIPS19)	one $\kappa < 0$		
κ -GCN (ICML20)	one $\kappa \in \mathbb{R}$		
LGCN (WWW21)	one $\kappa < 0$		
H-to-H (CVPR21)	one $\kappa < 0$		✓
SelfMG (AAAI22)	one $\kappa \in \mathbb{R}$		
fullyH (ACL22)	one $\kappa < 0$		✓
Q-GCN (NIPS22)	one radius $\frac{1}{ \kappa }$		
Ours	<i>diverse</i>	✓	✓

Table 1: Comparison of proposed MotifRGC and previous Riemannian methods. (κ denotes the curvature.)

The first issue is on the *curvature diversity*. Most of prior Riemannian methods (Zhang et al. 2021; Dai et al. 2021; Liu, Nickel, and Kiela 2019) study graphs in the manifold of a single curvature (radius), which is only suitable for a special type of graphs. For example, a negative curvature manifold (hyperbolic space) is well aligned with hierarchical and tree-like graphs (Nickel and Kiela 2017; Chen et al. 2022). A positive curvature manifold (hyperspherical space) is suitable for cyclical graphs (Bachmann, Bécigneul, and Ganea 2020). Note that, the product of single-curvature manifolds (Gu et al. 2019) still presents a single curvature as a whole. The recent quotient manifold (Xiong et al. 2022b) also relies on a single curvature radius. *There is a lack of curvature diversity to model the complex structures of real graphs.*

The second issue is on the *numerical stability*. Riemannian methods (Chami et al. 2019; Liu, Nickel, and Kiela 2019; Xiong et al. 2022b) typically model graphs with the couple of exponential and logarithmic maps. Unfortunately, exponential/logarithmic map is not strictly numerical-stable, and the issue of “Not a Number” (NaN) may occur in practice without numerical processing or careful hyperparameter tuning (Chen et al. 2022; Yu and Sa 2023). Recently, Dai et al. (2021) introduce the Lorentz-type operators to replace exponential/logarithmic map, but pose difficulty in optimization as side effect. Thus, *the issue of numerical stability largely remains open.*

The third issue is on the *motif regularity*. The motifs (small substructures such as triangles and cliques) are the fundamental building blocks of a graph, and they play an essential role in modeling and understanding social or bio-

chemical networks (Duval and Malliaros 2022; Yu and Gao 2022). Thus, encoding motif regularity is important in graph representation learning, and benefits downstream tasks, e.g., node classification and link prediction. In the literature, motif regularity has been extensively studied in Euclidean space (Liu and Sariyüce 2023; Subramonian 2021). Riemannian manifold tends to be more suitable to model motifs than the Euclidean counterpart, as will be shown in the experiment. Surprisingly, *it has been rarely explored in generic Riemannian manifolds, to our best knowledge.*

The aforementioned issues motivate us to rethink Riemannian graph representation learning, and propose a new problem of *Motif-aware Riemannian Graph Representation Learning*, which aims at finding a numerically stable graph encoder to model motif regularity in a novel diverse-curvature manifold. Besides, *self-supervised learning* without external labels is more practical, as labeling graphs is expensive or even impossible. In Riemannian manifolds, Yang et al. (2022); Sun et al. (2022b) leverage contrastive learning recently, while generative learning is still under investigated.

Our approach. To this end, we propose a novel Motif-aware Riemannian model with Generative-Contrastive Learning (**MotifRGC**), which conducts a minmax game in Riemannian manifold in a self-supervised manner. First, we propose a new type of GCN, namely Diverse-curvature GCN (D-GCN). In our design, we construct a diverse-curvature manifold by a product layer, where a *diversified factor* is introduced to ensure the curvature diversity (issue one). We replace the exponential/logarithmic map suffering from numerical stability (issue two) by a gyrovector *kernel layer*, where a numerical stable map based on Bochner’s Theorem is formulated. Second, we introduce the motif-aware Riemannian generative-contrastive learning, exploring the duality of the constructed manifold. In the product manifold, we design a *Riemannian motif generator* to generate fake motifs. The curvatures are learned to capture motif regularity (issue three), when the generated motifs become indistinguishable to the discriminator. Among the factor manifolds, node representations are learned by contrasting different geometric views of the factors, in which we introduce a *motif-aware hardness* to highlight the hard samples. We compare our model with prior methods in Table 1.

Notable contributions are summarized as follows:

- *Problem.* We make the first attempt to study motif-aware Riemannian graph representation learning, encoding motif regularity in a diverse-curvature manifold.
- *Methodology.* In MotifRGC, we propose a novel D-GCN coupled with several theoretical guarantees. Furthermore, we introduce the motif-aware Riemannian generative-contrastive learning to generate motif-aware node representations in a self-supervised manner.
- *Experiment.* Empirical results show MotifRGC outperforms previous Riemannian models. Codes are given in <https://github.com/RiemannGraph/MotifRGC>.

Preliminaries

We introduce necessary mathematics, specify the limitations of prior methods and propose the studied problem.

Riemannian Geometry

Manifold A Riemannian manifold \mathbb{M} is a smooth manifold coupled with a Riemannian metric. For each point \mathbf{x} , the Riemannian metric $g_{\mathbf{x}}$ is defined on its tangent space $\mathcal{T}_{\mathbf{x}}\mathbb{M}$. The logarithmic map $\text{Log}_{\mathbf{x}} : \mathbb{M} \rightarrow \mathcal{T}_{\mathbf{x}}\mathbb{M}$ transforms the vector in the manifold to the tangent space, while the exponential map $\text{Exp}_{\mathbf{x}}$ does the inverse transform. Euclidean space is a special case of Riemannian manifold.

Curvature Curvature measures the extent how a surface derives from being flat, and determines the shape of manifold. Each point \mathbf{x} in the manifold is associated with a curvature $\kappa_{\mathbf{x}}$ and a corresponding curvature radius $\frac{1}{|\kappa_{\mathbf{x}}|}$. A manifold is said to be a *single-curvature manifold* when the curvature (radius) of each point is equal. Specifically, the manifold is hyperbolic \mathbb{H} if the single curvature is negative, while the manifold is hyperspherical \mathbb{S} for a positive curvature. On the contrary, *diverse-curvature manifold* refers to a manifold where the curvatures of its points are not the same. Most previous works study graph in a single-curvature manifold (Yang et al. 2023a; Chami et al. 2019; Xiong et al. 2022b). **In fact, it calls for a diverse-curvature manifold, better matching the complex structures of real graphs.**

Euclidean & Riemannian GCNs

Graph Convolutional Networks (GCNs) are the dominant method for graph representation learning. GCNs typically conduct message passing over graphs (Velickovic et al. 2018; Hamilton, Ying, and Leskovec 2017). Concretely, in the convolution layer, each node representation \mathbf{h}_i aggregates the information of neighboring nodes \mathcal{N}_i and combines the aggregated information to itself. In Euclidean space, it takes the form of

$$\bar{\mathbf{h}}_i = \text{Agg}_{j \in \mathcal{N}_i}(\mathbf{h}_j), \quad \mathbf{h}_i = \text{Comb}(\bar{\mathbf{h}}_i, \mathbf{h}_i), \quad (1)$$

where Agg and Comb denote aggregation and combination, respectively. Similarly, a Riemannian GCN layer is generally formulated as follows,

$$\bar{\mathbf{h}}_i = \text{Agg}_{j \in \mathcal{N}_i}(\text{Log}_{\mathbf{0}}(\mathbf{h}_j)), \quad \mathbf{h}_i = \text{Exp}_{\mathbf{0}}(\text{Comb}(\bar{\mathbf{h}}_i, \mathbf{h}_i)), \quad (2)$$

where $\mathbf{0}$ denotes the reference point of exponential and logarithmic maps. However, as shown in Chen et al. (2022); Yu and Sa (2023), **the exponential/logarithmic map poses the issue of numerical stability.**

Problem Formulation

Notations. Lowercase \mathbf{x} and italic uppercase \mathcal{X} denote vector and set, respective. $\|\cdot\|$ denotes the usual $L2$ norm.

A graph is described as a tuple of $G(\mathcal{V}, \mathcal{E}, \mathcal{M}, \mathbf{X})$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the node set and $\mathcal{E} = \{(v_i, v_j)\} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the feature matrix, whose i^{th} row is the Euclidean feature of node v_i . Motifs $\mathcal{M} = \{M^3, \dots, M^K\}$ are small subgraphs frequently occurred in the graph, such as triangles and cliques. A motif M^K is the set of K -node connected subgraphs, where motif node set is $\mathcal{V}^K \subset \mathcal{V}$, $|\mathcal{V}^K| = K$, and motif edge set is $\mathcal{E}^K \subset \mathcal{E}$. We observe that **the motifs have not yet been explored in Riemannian manifold of diverse curvatures.** Also, we notice that labeling graph is often infeasible in practice. Thus, we propose a new problem as follows.

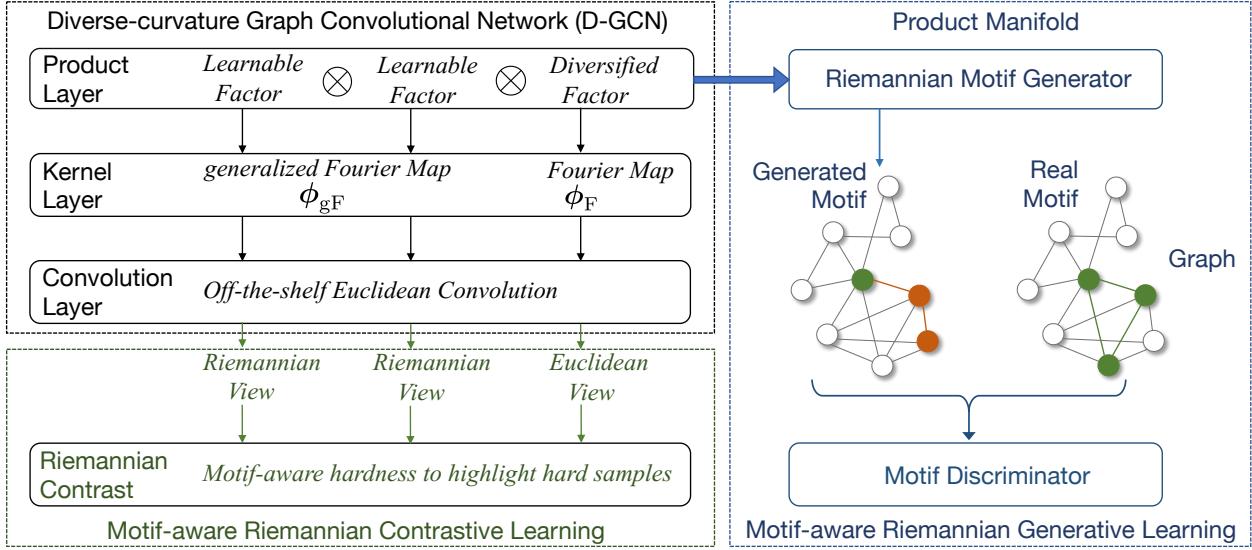


Figure 1: Overall architecture of MotifRGC. The proposed D-GCN consists of product layer, kernel layer and convolution layer. We learn the curvatures of the product manifold in the motif-aware Riemannian generative learning (blue box), and learn the node representations in the motif-aware Riemannian contrastive learning (olive box).

Problem Definition. (Motif-aware Riemannian Graph Representation Learning.) *Given a graph $G(\mathcal{V}, \mathcal{E}, \mathcal{M}, \mathbf{X})$ without any external labels, the proposed problem aims to learn a numerically stable encoding function $\Phi : \mathcal{V} \rightarrow \mathbb{M}$, so that motif regularity is captured in a diverse-curvature manifold \mathbb{M} , matching the complex structures of G .*

In short, we rethink Riemannian graph representation learning, and our work is distinguished from the prior works on curvature diversity, motif regularity and numerical stability.

Our Approach

To address this problem, we propose a novel self-supervised Motif-aware Riemannian model with Generative and Contrastive learning (**MotifRGC**). In a nutshell, we first propose a *new type of Riemannian GCN* to address the issues of curvature diversity and numerical stability, namely D-GCN. Then, we propose *Motif-aware Riemannian Generative-Contrastive Learning* to capture motif regularity via a min-max game in Riemannian manifold. The overall architecture is illustrated in Figure 1, and we elaborate on our D-GCN and learning approach in Sec 3.1 and Sec 3.2, respectively.

A New Formulation of D-GCN

Different from typical Riemannian GCNs in Eq. (2), we propose a novel Diverse-curvature GCN (termed as D-GCN), where we construct a diverse-curvature manifold via a product layer with a diversified factor, and replace the exponential/logarithmic map via a kernel layer. Another novelty lies in that the curvature learning of the manifold is conducted in a motif generation process, detailed in Sec. 3.2.

Product Layer The product layer conducts Cartesian product of multiple *learnable* single-curvature factors and

one diversified factor. A **single-curvature factor** is a Riemannian manifold of κ -stereographical model \mathbb{G}_κ^d (Petersen 2016), where κ and d denote curvature and dimension, respectively. Specifically, \mathbb{G}_κ^d is defined on a smooth gyrovector ball $\{\mathbf{x} \in \mathbb{R}^d \mid -\kappa \|\mathbf{x}\|^2 < 1\}$ with distance metric of $d(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{|\kappa|}} \tan_\kappa^{-1} \left(\sqrt{|\kappa|} \|\mathbf{x} \oplus_\kappa \mathbf{y}\| \right)$, where gyrovector addition \oplus_κ and curvature trigonometry \tan_κ^{-1} are detailed in the Appendix. The merit of κ -stereographical model is that it unifies hyperbolic and hyperspherical spaces in gyrovector formalism. The **diversified factor** is an upper hypersphere $\mathbb{S}_{\text{up}}^{d_0}$, which is responsible to diversify curvature of the product manifold. $\mathbb{S}_{\text{up}}^{d_0}$ is expressed in the polar coordinates of (r, θ) , where r and θ denote the magnitude coordinate and angular vector, respectively. Only r -coordinate contributes to curvature and distance, detailed in Appendix, and we set the norm of Euclidean feature as r .

The resulting product manifold is derived as follows

$$\mathbb{M} = \mathbb{G}_{\kappa_1}^{d_1} \otimes \mathbb{G}_{\kappa_2}^{d_2} \otimes \dots \otimes \mathbb{G}_{\kappa_M}^{d_M} \otimes \mathbb{S}_{\text{up}}^{d_0}, \quad (3)$$

where \otimes denotes the Cartesian product. A point in the product manifold takes the form $\mathbf{x} = [\mathbf{x}^1 \parallel \dots \parallel \mathbf{x}^M \parallel \mathbf{x}^0] \in \mathbb{M}$, where \parallel is vector concatenation. $\mathbf{x}^m \in \mathbb{G}_{\kappa_m}^{d_m}$ is the m^{th} component of \mathbf{x} , and $\mathbf{x}^0 \in \mathbb{S}_{\text{up}}^{d_0}$. The distance metric is

$$d_{\mathbb{M}}^2(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^M d_{\kappa_m}^2(\mathbf{x}^m, \mathbf{y}^m) + (\|\mathbf{x}^0\| - \|\mathbf{y}^0\|)^2. \quad (4)$$

With Eqs. (3)–(4), we have Proposition 1 hold.

Proposition 1 (Curvature Diversity). *The manifold \mathbb{M} constructed by the product layer (Eq. 3) is a diverse-curvature manifold in which each point has its own curvature $\kappa_{\mathbf{x}}$ determined by its location in the manifold.*

Proof. Please refer to the Appendix. □

Kernel Layer Recall that, typical Riemannian GCN layer with exponential/logarithmic map suffers from the issue of numerical stability. Instead, we put forward a fresh perspective of kernel method: We transform Riemannian features to Euclidean ones via a *gyrovector kernel*, and then utilize well-established Euclidean GCN layers to update node representations, as shown in Fig. 1.

The transformation between two types of spaces is challenging due to the difference in geometry and constraint on gyrovector ball. In kernel layer, we formulate a **generalized Fourier Map** $\phi_{\text{gF}} : \mathbb{G}_{\kappa}^n \rightarrow \mathbb{R}^m$ to address this challenge. *First, we start with Fourier (feature) map* $\phi_{\text{F}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in Euclidean space. The Bochner’s Theorem (Rahimi and Recht 2007) utilizes eigenfunctions to construct the Fourier map for any invariant kernel. A typical Euclidean eigenfunction family takes the form of

$$F_{\omega,b}(\mathbf{x}) = \sqrt{2} \cos(\langle \omega, \mathbf{x} \rangle + b), \mathbf{x} \in \mathbb{R}^n, \quad (5)$$

where phase vector ω and bias b are uniformly sampled from a n -dimensional unit ball and $[0, 2\pi]$, respectively.

Second, we generalize Euclidean $F_{\omega,b}(\cdot)$ to Riemannian manifold. Analogous to Eq. (5), we derive the eigenfunction in the gyrovector ball \mathbb{G}_{κ}^n of Riemannian manifold,

$$\text{gF}_{\omega,b,\lambda}^{\kappa}(\mathbf{x}) = A_{\omega,\mathbf{x}} \cos(\lambda \langle \omega, \mathbf{x} \rangle_{\kappa} + b), \mathbf{x} \in \mathbb{G}_{\kappa}^n, \quad (6)$$

where the amplitude function $A_{\omega,\mathbf{x}}$ is $\exp\left(\frac{n-1}{2} \langle \omega, \mathbf{x} \rangle_{\kappa}\right)$. $\langle \omega, \mathbf{x} \rangle_{\kappa} = \log \frac{1+\kappa \|\mathbf{x}\|^2}{\|\mathbf{x}-\omega\|^2}$ is the signed distance in the gyrovector ball. We draw m independent samples of ω , b and λ uniformly from n -dimensional gyrovector ball, $[0, 2\pi]$ and Gaussian distribution, respectively. The generalized Fourier map is then given as

$$\phi_{\text{gF}}(\mathbf{x}) = \frac{1}{\sqrt{m}} [\text{gF}_{\omega_1,\lambda_1,b_1}^{\kappa}(\mathbf{x}), \dots, \text{gF}_{\omega_m,\lambda_m,b_m}^{\kappa}(\mathbf{x})] \in \mathbb{R}^m. \quad (7)$$

Theoretically, we show that induced kernel can be regarded as a generalization of Poisson kernel (Sonoda, Ishikawa, and Ikeda 2022) to Riemannian manifold.

Proposition 2 (Poisson Kernel). *Let the radius of gyrovector ball be 1 (curvature $\kappa = -1$), the kernel in Proposition 2 is equivalent to the famous Poisson kernel.*

Proof. Please refer to the Appendix. \square

Network Architecture In our D-GCN, the product layer assigns each node with multiple learnable Riemannian features, and then constructs the diverse-curvature manifold. The kernel layer conducts ϕ_{gF} to Riemannian features and ϕ_{F} to Euclidean features. The convolution layer can be given by any off-the-shelf Euclidean one.

Motif-aware Generative-Contrastive Learning on Riemannian Manifold

We propose a novel Motif-aware Riemannian Generative-Contrastive learning, in which *we introduce a minmax game to Riemannian manifolds.* Thanks to the formulation in Eq. (3), the constructed manifold presents a duality of the product manifold as a whole and the collaboration of factor manifolds. Exploring the duality of the constructed manifold, we

conduct motif generation process in the product manifold for the curvature learning (*Generative Learning*). Simultaneously, we contrast among factor manifolds for the representation learning (*Contrastive Learning*), where a motif-aware hardness is introduced.

In generative learning, the product manifold is trained to capture motif regularity, so that *fake motifs \mathcal{S} generated from the generator G cannot be distinguished from real motifs \mathcal{M} by the discriminator D .* Specially, generator G and discriminator D play the minmax game as follows,

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{\mathcal{M}}[\log D(\mathcal{M}; \theta_D)] + \mathbb{E}_{\mathcal{S} \sim G(\mathcal{S}; \theta_G)}[\log(1 - D(\mathcal{S}; \theta_D))]. \quad (8)$$

Riemannian Motif Generator We design a Riemannian motif generator G to generate indistinguishable motifs in the minimization of Eq. (8). A motif \mathcal{M}^k is a set of k nodes $\mathcal{S} = \{v_{s_1}, \dots, v_{s_k}\}$. Let \mathcal{S}_{s_1} denote the node set including v_{s_1} . Given v_{s_1} , the selection of other $(k-1)$ nodes is guided by the proposed generator defined as follows,

$$G(\mathcal{S}_{s_1} | v_{s_1}) = G_c(v_{s_2} | v_{s_1}, v_{s_2}) \cdots G_c(v_{s_k} | v_{s_1}, \dots, v_{s_{k-1}}) \quad (9)$$

The generation of v_{s_k} is based on previously selected nodes $\mathcal{S}' = \{v_{s_1}, \dots, v_{s_{k-1}}\}$, and the probability of v_{s_k} being selected is defined by **distance metric in the product \mathbb{M}** . Concretely, we apply a softmax function over all other nodes,

$$G_c(v_{s_k} | v_{s_1}, \dots, v_{s_{k-1}}) = \frac{\exp(d_{\mathbb{M}}(v_{s_k}, \mu_{s_1, \dots, s_{k-1}}))}{\sum_{v_{s_j} \notin \mathcal{S}'} \exp(d_{\mathbb{M}}(v_{s_j}, \mu_{s_1, \dots, s_{k-1}}))}, \quad (10)$$

where $d_{\mathbb{M}}$ given in Eq. (4) is a function regarding the curvatures in the product. $\mu_{s_1, \dots, s_{k-1}}$ is the geometric centroid of the nodes \mathcal{S}' in \mathbb{M} . Thanks to combinational formulation of $d_{\mathbb{M}}$, there is no need to consider the centroid as a whole, and only the centroid in each factor $\mu_{s_1, \dots, s_{k-1}}^m$ is required.

For each factor of gyrovector ball $\mathbb{G}_{\kappa_m}^{d_m}$, it is given by the gyro-midpoint (Bachmann, Bécigneul, and Ganea 2020),

$$\mu_{s_1, \dots, s_{k-1}}^m = \sum_{i=1}^{k-1} \frac{\lambda_{v_{s_i}}^{\kappa_m}}{\sum_{j=1}^{k-1} (\lambda_{v_{s_j}}^{\kappa_m} - 1)} v_{s_i}^m \in \mathbb{G}_{\kappa_m}^{d_m}, \quad (11)$$

where $\lambda_v^{\kappa} = 2/(1 + \kappa \|v\|^2)$ is the conformal factor.

In the minimization of Eq. (8), curvatures of the product are the parameters θ_G to be learned. The intuition is that *the product manifold is able to generate indistinguishable motifs, when its curvatures (and corresponding distance metric) capture the motif regularity in graph.* The sampling of motifs $\mathcal{S} \sim G(\mathcal{S}; \theta_G)$ is discrete, and thus we minimize Eq. (8) via the policy gradient (Wang et al. 2018) to train θ_G . Here, we focus on a fundamental motif of triangle.

Motif Discriminator In the maximization of Eq. (8), the discriminator D aims to classify the generated motifs from the real ones sampled from the graph. Given a node set $\{v_{s_1}, \dots, v_{s_k}\}$, we implement the discriminator as follows,

$$D(v_{s_1}, \dots, v_{s_k} | \theta_D) = \text{MLP}(\text{Pooling}(v_{s_1}, \dots, v_{s_k})), \quad (12)$$

where MLP is a MultiLayer Perception whose output layer is the sigmoid. We apply mean-pooling to the node representations from the kernel layer of our Riemannian GCN, so

that the representations can be tackled by a normal MLP. Thus, θ_D is the parameter of MLP.

Motif-aware Riemannian Contrast We formulate a minimization objective for motif-aware Riemannian contrastive learning, in which we introduce a motif-aware hardness to highlight hard samples for learning node representations.

First, we generate multiple geometric views, and each factor manifold provides a geometric view of corresponding geometry. For factor manifold $\mathbb{G}_{\kappa^m}^{d_m}$, its geometric view is derived as $\mathbf{z}^m = g_{\Theta}(\phi_{g_F}(\mathbf{x}^m))$, and $\mathbf{z}^0 = g_{\Theta}(\phi_F(\mathbf{x}^0))$ for Euclidean view. \mathbf{x}^m and \mathbf{x}^0 denote the Riemannian features and Euclidean features, respectively. Each geometric view is contrasted to Euclidean view and vice versa, so that positive samples are close and negative samples are pushed away.

Second, we leverage the motif to distinguish positive/negative samples. Intuitively, the nodes in a motif are similar to each other, and are considered as positive samples. Take the motif of triangle \mathcal{M}^3 for instance. For node v_i , the set of positive samples is denoted as $\mathcal{V}^3(i)$, which is a collection of the endpoints in triangles including node v_i .

Furthermore, we formulate a **motif-aware hardness** to highlight the hard samples, i.e., the positive samples with low similarity and the negative samples with high similarity.

$$h(\mathbf{z}_i, \mathbf{z}_j | \mathcal{M}_3) = |\mathbb{I}(j \in \mathcal{V}^3(i)) - \text{Normal}(s(\mathbf{z}_i, \mathbf{z}_j))|^\alpha, \quad (13)$$

where the indicator function $\mathbb{I}(\cdot)$ returns 1 iff (\cdot) is true, s is a similarity measure, and Normal normalizes the similarity value to $[0, 1]$. The positive α controls the effect of hardness. Eq. (13) up-weights the hard samples. For example, given $\alpha = 2$, a hard negative sample with similarity $s = 0.9$ is reweighed by 0.81, while an easy negative with $s = 0.1$ is reweighed by 0.01. The reweighing of motif-aware hardness pays more attention to the hard samples than the typical equal treatment, i.e., InfoNCE loss (Oord, Li, and Vinyals 2018). Our insight is to inject motif-aware hardness, and thus the minimization of Riemannian Contrast (RC) is

$$\text{RC}_m^0 = - \sum_{i=1}^N \log \frac{\exp(s(\mathbf{z}_i^m, \mathbf{z}_i^0))}{\sum_{j=1}^N \exp(h(\mathbf{z}_i^m, \mathbf{z}_j^0 | \mathcal{M}_3) s(\mathbf{z}_i^m, \mathbf{z}_j^0))}. \quad (14)$$

Note that, Eq. (14) recovers the InfoNEC with $\alpha = 0$.

Overall Minmax Objective. The overall minimization is given by incorporating the objective of Riemannian motif generator and motif-aware Riemannian contrast,

$$\min \mathbb{E}_{\mathcal{S} \sim G(\mathcal{S}; \theta_G)} [\log(1 - D(\mathcal{S}; \theta_D))] + \sum_{m=1}^M (\text{RC}_m^0 + \text{RC}_m^m). \quad (15)$$

while maximization is the objective of the discriminator,

$$\max \mathbb{E}_{\mathcal{M}} [\log D(\mathcal{M}; \theta_D)] + \mathbb{E}_{\mathcal{S} \sim G(\mathcal{S}; \theta_G)} [\log(1 - D(\mathcal{S}; \theta_D))]. \quad (16)$$

The overall process is summarized in Algorithm 1, where the minimization is given in Lines 4-8 while maximization in Lines 10-12. Line 1 is specified in Experiment. *Alternatively optimizing the minimization in Eq. 15 and maximization in Eq. 16, we learn node representations in the diverse-curvature manifold, where motif regularity is captured.*

Algorithm 1: Optimizing MotifRGC

Input: Graph $G(\mathcal{V}, \mathcal{E}, \mathcal{M}, \mathbf{X})$
 $\text{MinSteps}, \text{MaxSteps}$

Output: Riemannian graph encoder Φ
 Motif generator G
 The discriminator D

```

1 Initialize  $\Phi, G, D$  and Riemannian feature  $\mathbf{x}^m$ ;
2 while not converging do
3    $\triangleright$  Riemannian Generative-Contrastive Learning
4   for MinSteps do
5     Generate multiple geometric views;
6     Calculate motif-aware hardness in Eq. 13;
7     Generate fake motifs from  $G$  with Eqs. 9-10;
8     Update  $\Phi$  and  $G$  according to Eq. 15;
9    $\triangleright$  Training the Discriminator
10  for MaxSteps do
11    Sample real motifs and fake motifs generated
12    from  $G$ ;
13    Update  $D$  according to Eq. 16;

```

Experiment

Datasets & Baselines We choose 4 public datasets: Cora, Citeseer and Pubmed (Yang, Cohen, and Salakhutdinov 2016), and Airport (Chami et al. 2019). We include 8 supervised baselines: HGNN (Liu, Nickel, and Kiela 2019), HGCN (Chami et al. 2019), LGCN (Zhang et al. 2021), κ -GCN (Bachmann, Bécigneul, and Ganea 2020), H-to-H (Dai et al. 2021), fullyH (Chen et al. 2022) and \mathcal{Q} -GCN (Xiong et al. 2022b), and a self-supervised SelfMG (Sun et al. 2022b). We implement κ -GCN in the product manifold, showing the result of the product without our diversified factor. *Few study considers motif in Riemannian manifold to our best knowledge, and we bridge this gap in this paper.* We implement **our model** with the backbone of GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018) and SAGE (Hamilton, Ying, and Leskovec 2017), and thus we list the results of the 3 backbones. In addition, we include the recent SME (Jiang et al. 2022) as a baseline, which considers the motif in Euclidean space.

Evaluation Metrics We examine our model in both link prediction and node classification. The evaluation metrics for link prediction is AUC and Average Precision (AP) (Chami et al. 2019), while for node classification, we employ Accuracy (ACC) (Kipf and Welling 2017).

Reproducibility In our model, the convolution layer is stacked twice, and MLP has 2 hidden layers. The number of learnable factors is 3 with the curvatures $\kappa_1 = 1$, $\kappa_2 = -1$ and $\kappa_3 = -1$ as default. $\alpha = 2$ for contrastive learning. To initialize Riemannian features, we first initialize $\mathbf{X} \in \mathbb{R}^{N \times d_m}$, where N and d_m are number of nodes and factor dimension. Then, we have $\mathbf{X}^m = \frac{\mathbf{X}}{2\sqrt{\kappa_m} \|\mathbf{X}\|_{\max}} \in \mathbb{G}_{\kappa_m}^{d_m}$ in the factor manifold, where $\|\mathbf{X}\|_{\max}$ is the maximum norm of the rows. Riemannian features are optimized by Riemannian Adam (Bécigneul and Ganea 2019), while others are optimized by Adam (Kingma and Ba 2015).

Method	Cora		Citeseer		Pubmed		Airport	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
GCN	90.11(0.51)	91.52(0.68)	90.16(0.49)	92.91(0.33)	91.16(0.36)	91.96(1.27)	89.29(0.38)	91.37(0.29)
GAT	92.55(0.49)	93.41(1.01)	89.32(0.36)	92.20(1.51)	91.21(0.24)	92.07(1.05)	91.42(1.20)	92.26(1.06)
SAGE	86.02(0.55)	90.20(0.76)	88.18(0.22)	89.07(0.18)	87.93(0.15)	91.42(0.12)	89.75(0.67)	92.12(0.74)
SME	92.13(0.28)	93.24(0.20)	93.20(0.66)	94.56(0.92)	93.11(0.09)	93.36(0.75)	92.68(0.33)	94.08(0.57)
HGCN	93.60(0.37)	94.13(0.27)	94.33(0.42)	94.90(0.24)	95.43(0.02)	95.44(0.02)	95.11(0.26)	95.09(0.31)
HGNN	91.59(1.67)	91.34(1.37)	96.05(0.42)	96.82(0.44)	94.37(0.63)	93.57(1.03)	92.46(0.39)	93.36(0.82)
LGCN	92.69(0.26)	93.37(0.26)	93.49(1.11)	94.32(0.77)	95.40(0.20)	95.52(0.22)	95.50(0.06)	95.61(0.11)
κ -GNN	94.25(1.23)	93.62(1.76)	97.06(0.67)	96.59(0.43)	94.90(0.30)	94.84(0.13)	95.08(0.81)	95.21(0.80)
H-to-H	89.71(0.85)	91.70(0.45)	91.23(3.13)	92.64(2.18)	97.10(0.03)	96.57(0.07)	96.91(0.03)	95.59(0.03)
SelfMG	94.28(0.64)	93.95(0.72)	96.52(0.36)	95.77(1.52)	97.32(0.24)	95.16(0.53)	96.85(0.53)	94.33(1.05)
Q -GCN	95.22(0.29)	97.09(2.03)	94.31(0.73)	94.85(0.49)	94.69(0.18)	95.87(0.62)	96.49(0.13)	97.01(1.12)
fullyH	91.81(1.92)	92.20(0.61)	91.42(0.28)	91.33(0.81)	97.51(3.89)	92.55(2.71)	94.77(0.41)	94.02(0.11)
Ours _{GCN}	97.37(2.06)	97.95(1.02)	98.54(0.67)	97.46(0.15)	98.96(0.36)	98.84(0.40)	97.22(0.42)	97.28(0.27)
Ours _{GAT}	98.86 (1.02)	98.64 (0.15)	98.85 (0.82)	99.02 (0.75)	99.09 (0.13)	98.93 (0.18)	97.40 (0.52)	96.72 (0.59)
Ours _{SAGE}	98.19 (0.79)	99.02 (0.88)	98.99 (0.51)	99.12 (0.52)	98.92(0.67)	98.69(0.78)	97.90 (0.35)	97.62 (0.40)

Table 2: Link prediction results on Cora, Citeseer, Pubmed and Airport datasets in terms of AUC (%) and AP (%). Standard derivations are given in the brackets. The best results are boldfaced and the runner up underlined.

Method	Cora	Citeseer	Pubmed	Airport
GCN	81.1(0.3)	70.2(0.2)	77.7(1.0)	81.7(0.4)
GAT	81.9(1.1)	71.3(1.4)	78.6(1.3)	82.1(1.2)
SAGE	77.9(1.7)	69.3(1.9)	77.2(0.5)	82.1(0.9)
SME	81.5(0.6)	72.5(1.0)	78.4(0.7)	83.9(0.7)
HGCN	81.1(0.2)	72.5(1.0)	80.1(1.6)	89.2(0.6)
HGNN	80.8(0.6)	71.1(2.2)	79.7(1.2)	82.7(2.2)
LGCN	<u>82.6</u> (0.3)	72.4(0.3)	80.3(0.3)	89.2(1.3)
κ -GCN	81.2(0.4)	73.1(0.6)	81.2(0.5)	84.8(1.5)
H-to-H	80.0(0.2)	71.0(0.3)	79.5(0.3)	81.8(0.9)
SelfMG	82.2(1.0)	73.2(1.0)	80.0(1.2)	86.1(1.2)
Q -GCN	81.7(0.6)	73.5(0.4)	80.9(0.8)	83.2(0.8)
fullyH	79.9(0.4)	71.9(1.1)	79.9(0.5)	86.4(0.3)
Ours _{GCN}	83.7 (1.2)	74.2 (1.1)	82.2 (1.1)	89.2(1.8)
Ours _{GAT}	82.5(0.7)	<u>73.9</u> (0.7)	81.5(0.9)	91.5 (1.3)
Ours _{SAGE}	82.0(0.5)	73.3(1.1)	<u>81.6</u> (1.0)	<u>90.9</u> (1.8)

Table 3: Node classification results in terms of ACC (%).

Empirical Results

Link prediction & Node classification We summarize the results for link prediction and node classification in Table 2 and Table 3, respectively. We run each method 10 times independently, and report the mean value with standard deviation for fair comparisons. For our model, the representation dimension is 128 with 32 for each factor, except two factors on Pubmed. All the methods leverage Fermi-Dirac decoder (Nickel and Kiela 2017) for link prediction, where the distance is given by the respective geometry. We achieve the best results among 12 baselines, e.g., our model of GCN backbone achieves at least 7.36% AUC gain to the backbone itself. The reason is that the constructed manifold better matches graph structures, and our design for motifs benefits representation and curvature learning.

On Numerical Stability Typical Riemannian GCNs with the exponential/logarithmic map sometimes encounter “Not a Number (NaN)” issue in PyTorch, as will be shown in

Variant	Cora		Citeseer		
	LP	NC	LP	NC	
GCN	MotifRGC	97.37	83.67	98.54	74.24
	– Hard	<u>97.05</u>	<u>83.12</u>	<u>98.36</u>	<u>73.99</u>
	– RCL	96.30	82.34	97.86	71.98
	– gF	96.92	82.80	NaN	73.20
	– Motif	95.02	81.29	96.80	72.16
GAT	MotifRGC	98.86	82.55	98.85	73.95
	– Hard	<u>98.45</u>	<u>82.20</u>	<u>98.27</u>	<u>73.39</u>
	– RCL	97.99	81.06	97.86	72.93
	– gF	98.10	82.02	<u>98.79</u>	73.08
	– Motif	95.18	82.13	97.11	72.67
SAGE	MotifRGC	98.19	82.02	98.99	73.35
	– Hard	<u>98.02</u>	<u>81.88</u>	98.72	<u>72.86</u>
	– RCL	96.81	79.02	97.15	72.11
	– gF	97.96	81.30	<u>98.88</u>	72.57
	– Motif	94.50	81.49	96.67	72.32

Table 4: Ablation Study on Proposed Components. AUC (%) for Link Prediction (LP). ACC(%) for Node Classification (NC). Best result are boldfaced and runner-up underlined.

Ablation Study. To avoid this issue, κ -GNN carefully tunes the curvatures for the maps. HGCN, LGCN, SelfMG and Q -GCN employ the post-process with Normalize function, forcing the representations to fit Riemannian manifolds. H-to-H alternatively introduce Lorentz-type operators but the sophisticated optimization in Stiefel manifold is nontrivial. On the contrary, the proposed gF is numerical stable.

Ablation Study Here, we evaluate the effectiveness of each proposed component of MotifRGC: 1) generalized Fourier map gF, 2) Riemannian motif generator, 3) Riemannian contrastive learning, 4) motif-aware hardness as well as the number of learnable factors.

To this end, we introduce a variety of variant models. For – gF variant, we replace gF with logarithmic map to evaluate the numerical stability of gF. For – motif variant, we

Variant		Cora		Citeseer	
		LP	NC	LP	NC
GCN	$\mathbb{G} \times \mathbb{S}_{\text{up}}$	95.27	81.30	96.33	71.92
	$(\mathbb{G})^2 \times \mathbb{S}_{\text{up}}$	95.80	82.06	97.46	73.14
	$(\mathbb{G})^3 \times \mathbb{S}_{\text{up}}$	97.37	83.67	98.54	74.24
	$(\mathbb{G})^4 \times \mathbb{S}_{\text{up}}$	97.99	83.83	99.03	74.18
GAT	$\mathbb{G} \times \mathbb{S}_{\text{up}}$	95.87	82.09	97.68	72.60
	$(\mathbb{G})^2 \times \mathbb{S}_{\text{up}}$	97.58	82.24	98.50	73.31
	$(\mathbb{G})^3 \times \mathbb{S}_{\text{up}}$	<u>98.86</u>	82.55	<u>98.85</u>	<u>73.95</u>
	$(\mathbb{G})^4 \times \mathbb{S}_{\text{up}}$	99.16	82.40	98.96	74.02

Table 5: Ablation Study on Learnable Factors.

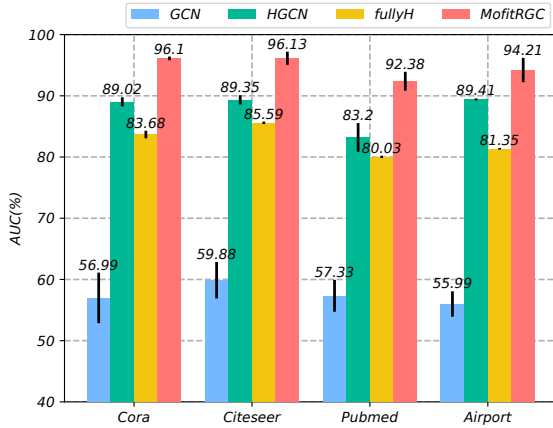


Figure 2: AUC (%) of Triangle Generation/Prediction.

train our model with the contrastive loss only, to evaluate the importance of motif generation. For $-$ RCL variant, we disable Riemannian contrastive learning in the minmax objective. For $-$ Hard variant, we replace RCL with InfoNCE loss (Oord, Li, and Vinyals 2018) in the minmax objective to evaluate the motif-aware hardness in contrastive learning. The variants are instantiated on the product of three learnable factors by default, i.e., $(\mathbb{G})^3 \times \mathbb{S}_{\text{up}}$. We record the results for both tasks on Cora and Citeseer datasets in Table 4, where the derivation is given in brackets. It shows that: 1) $-$ gF variant has inferior performance to our model. The reason lies in that the logarithmic map suffers from numerical instability (Chen et al. 2022; Yu and Sa 2023). The issue of “NaN” may occur, as shown in the link prediction on Citeseer, *motivating our formulation of gF*. 2) RCL increases the performance of our model, and motif-aware hardness is important for RCL. 3) Our model consistently outperforms the $-$ motif variant. *It verifies our insight: motif regularity is effective to learn the curvatures*, offering a new perspective to curvature learning in Riemannian manifold. On the number of learnable factors, we instantiate our model with different numbers of factors, and record the results in Table 5. It suggests: the product with more factor is more flexible to match the graph structures in general, achieving better results.

Case Study on Triangle Generation We conduct the case

study to examine triangle generation from the proposed MotifRGC. Concretely, MotifRGC is instantiated on the backbone of GCN, and the generated triangles are evaluated by AUC metric given real triangles. Previous Riemannian methods lack the ability of motif generation, to our best knowledge. Instead, we list the AUC of triangle prediction, where a triangle is predicted if all the three edges are correctly predicted. Fig. 2 shows the result of triangle generation/prediction on Cora, Citeseer, Pubmed and Airport datasets. Standard deviation is given in the error bar. In Fig. 2, Riemannian models outperform the Euclidean GCN. It suggests that Riemannian manifold tends to be more suitable to model motifs than the Euclidean counterpart. Our MotifRGC achieves the best result, with at least 4.8% AUC gain to the runnerup and 35.05% AUC gain to its backbone. It shows *the learnt manifold captures the motif regularity of real graphs, and thus is able to generate triangles effectively*.

Related Work

On Riemannian graph learning, Nickel and Kiela (2017); Suzuki, Takahama, and Onoda (2019) introduce hyperbolic space to model graphs. Liu, Nickel, and Kiela (2019); Chami et al. (2019); Zhang et al. (2021); Li et al. (2022b) generalize GNN to hyperbolic space, respectively. Fu et al. (2021) propose to explore curvature in hyperbolic space. H-to-H (Dai et al. 2021) and *fullyH* (Chen et al. 2022) propose the Lorentz type operations. κ -GCN (Bachmann, Bécigneul, and Ganea 2020) further extends GCN to any curvature. \mathcal{Q} -GCN (Xiong et al. 2022b) introduces a quotient manifold. Yang et al. (2022); Zhu et al. (2020) embed graphs in the dual spaces of Euclidean and hyperbolic ones. Crueru, Bécigneul, and Ganea (2021); López et al. (2021) study Riemannian matrix spaces. Yang et al. (2023a); Liu et al. (2023); Nguyen et al. (2023) explore Ricci curvature on graphs. Xiong et al. (2022a); Wang et al. (2021) focus on knowledge graphs. SelfMG (Sun et al. 2022b) and κ -GCN also study graphs in the product space (Gu et al. 2019) but, as mentioned before, *we specify the product itself cannot result in diverse-curvature manifold*. Giovanni, Luise, and Bronstein (2022) study the curvature diversity of an upper hypersphere mathematically, but *neither build GCN nor consider curvature learning*. Note that, Sonoda, Ishikawa, and Ikeda (2022); Yu and Sa (2023) formulate kernels for hyperbolic space, while *we formulate a kernel for any Riemannian manifold*. Recently, Sun et al. (2023c, 2022a, 2021) propose Riemannian GNNs for temporal graphs. Sun et al. (2023a,b) rethink structure learning and clustering in the manifold.

Conclusion

We rethink Riemannian graph representation learning, and propose the first motif-aware model in diverse-curvature manifold. We design a new type of Riemannian GCN (D-GCN), where we ensure curvature diversity by the product layer, and address numerical stability by the kernel layer. Then, motif-aware Riemannian generative-contrastive learning introduces a minmax game in the constructed manifold, capturing motif-regularity in node representations. Extensive experiments show the superiority of our model.

Acknowledgments

This work is supported by National Key R&D Program of China through grant 2021YFB1714800, the National Natural Science Foundation of China through grant 62202164, and the Fundamental Research Funds for the Central Universities (2022MS018). Prof. Philip S. Yu is supported in part by NSF under grant III-2106758. Corresponding Authors: Li Sun and Hao Peng.

References

- Bachmann, G.; Bécigneul, G.; and Ganea, O. 2020. Constant Curvature Graph Convolutional Networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, 486–496. PMLR.
- Bécigneul, G.; and Ganea, O. 2019. Riemannian Adaptive Optimization Methods. In *Proceedings of 7th International Conference on Learning Representation (ICLR)*.
- Chami, I.; Ying, Z.; Ré, C.; and Leskovec, J. 2019. Hyperbolic Graph Convolutional Neural Networks. In *Advances in the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, 4869–4880.
- Chen, W.; Han, X.; Lin, Y.; Zhao, H.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2022. Fully Hyperbolic Neural Networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 5672–5686. ACL.
- Cruceru, C.; Bécigneul, G.; and Ganea, O. 2021. Computationally Tractable Riemannian Manifolds for Graph Embeddings. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 7133–7141. AAAI Press.
- Dai, J.; Wu, Y.; Gao, Z.; and Jia, Y. 2021. A Hyperbolic-to-Hyperbolic Graph Convolutional Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 154–163. CVF/ IEEE.
- Duval, A.; and Malliaros, F. D. 2022. Higher-order Clustering and Pooling for Graph Neural Networks. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, 426–435.
- Fu, X.; Li, J.; Wu, J.; Sun, Q.; Ji, C.; Wang, S.; Tan, J.; Peng, H.; and Yu, P. S. 2021. ACE-HGNN: Adaptive Curvature Exploration Hyperbolic Graph Neural Network. In *Proceedings of the 21st IEEE International Conference on Data Mining (ICDM)*, 111–120. IEEE.
- Giovanni, F. D.; Luise, G.; and Bronstein, M. M. 2022. Heterogeneous manifolds for curvature-aware graph embedding. In *Proceedings of the 10th International Conference on Learning Representation (GTRL Workshop)*.
- Gu, A.; Sala, F.; Gunel, B.; and Ré, C. 2019. Learning Mixed-Curvature Representations in Product Spaces. In *Proceedings of the 7th International Conference on Learning Representation (ICLR)*.
- Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in the 30th Conference on Neural Information Processing Systems (NeurIPS)*, 1024–1034.
- Jiang, X.; Yang, Z.; Wen, P.; Su, L.; and Huang, Q. 2022. A Sparse-Motif Ensemble Graph Convolutional Network against Over-smoothing. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2094–2100.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representation (ICLR)*.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representation (ICLR)*.
- Krioukov, D. V.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; and Boguñá, M. 2010. Hyperbolic Geometry of Complex Networks. *CoRR*, abs/1006.5169.
- Li, J.; Fu, X.; Sun, Q.; Ji, C.; Tan, J.; Wu, J.; and Peng, H. 2022a. Curvature Graph Generative Adversarial Networks. In *Proceedings of The ACM Web Conference 2022*, 1528–1537. ACM.
- Li, J.; Fu, X.; Sun, Q.; Ji, C.; Tan, J.; Wu, J.; and Peng, H. 2022b. Curvature Graph Generative Adversarial Networks. In *Proceedings of the ACM Web Conference 2022*, 1528–1537. ACM.
- Liu, P.; and Sariyüce, A. E. 2023. Using Motif Transitions for Temporal Graph Generation. In *Proceedings of the 29th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1501–1511. ACM.
- Liu, Q.; Nickel, M.; and Kiela, D. 2019. Hyperbolic Graph Neural Networks. In *Advances in the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, 8228–8239.
- Liu, Y.; Zhou, C.; Pan, S.; Wu, J.; Li, Z.; Chen, H.; and Zhang, P. 2023. CurvDrop: A Ricci Curvature Based Approach to Prevent Graph Neural Networks from Over-Smoothing and Over-Squashing. In *Proceedings of the ACM Web Conference 2023*, 221–230. ACM.
- López, F.; Pozzetti, B.; Trettel, S.; Strube, M.; and Wienhard, A. 2021. Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, 7090–7101. PMLR.
- Nguyen, K.; Nong, H.; Nguyen, V.; Ho, N.; Osher, S.; and Nguyen, T. 2023. Revisiting Over-smoothing and Over-squashing Using Ollivier-Ricci Curvature. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR.
- Nickel, M.; and Kiela, D. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in 30th Conference on Neural Information Processing Systems (NeurIPS)*, 6338–6347.
- Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748.
- Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online learning of social representations. In *Proceedings of the 20th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 701–710. ACM.

- Petersen, P. 2016. *Riemannian Geometry*, volume 171 of *Graduate Texts in Mathematics*. Springer International Publishing.
- Rahimi, A.; and Recht, B. 2007. Random Features for Large-Scale Kernel Machines. In *Advances in 20th Conference on Neural Information Processing Systems (NeurIPS)*, 1177–1184.
- Shimizu, R.; Mukuta, Y.; and Harada, T. 2021. Hyperbolic Neural Networks++. In *Proceedings of 9th International Conference on Learning Representation (ICLR)*.
- Sonoda, S.; Ishikawa, I.; and Ikeda, M. 2022. Fully-Connected Network on Noncompact Symmetric Space and Ridgelet Transform based on Helgason-Fourier Analysis. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, 20405–20422. PMLR.
- Subramonian, A. 2021. Motif-Driven Contrastive Learning of Graph Representations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 15980–15981. AAAI Press.
- Sun, L.; Huang, Z.; Wu, H.; Ye, J.; Peng, H.; Yu, Z.; and Yu, P. S. 2023a. DeepRicci: Self-supervised Graph Structure-Feature Co-Refinement for Alleviating Over-squashing. In *Proceedings of the 23rd IEEE International Conference on Data Mining (ICDM)*.
- Sun, L.; Wang, F.; Ye, J.; Peng, H.; and Yu, P. S. 2023b. Congregate: Contrastive Graph Clustering in Curvature Spaces. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2296–2305.
- Sun, L.; Ye, J.; Peng, H.; Wang, F.; and Yu, P. S. 2023c. Self-Supervised Continual Graph Learning in Adaptive Riemannian Spaces. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 4633–4642.
- Sun, L.; Ye, J.; Peng, H.; and Yu, P. S. 2022a. A Self-supervised Riemannian GNN with Time Varying Curvature for Temporal Graph Learning. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, 1827–1836. ACM.
- Sun, L.; Zhang, Z.; Ye, J.; Peng, H.; Zhang, J.; Su, S.; and Yu, P. S. 2022b. A Self-Supervised Mixed-Curvature Graph Neural Network. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 4146–4155. AAAI Press.
- Sun, L.; Zhang, Z.; Zhang, J.; Wang, F.; Peng, H.; Su, S.; and Yu, P. S. 2021. Hyperbolic Variational Graph Neural Network for Modeling Dynamic Graphs. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 4375–4383.
- Suzuki, R.; Takahama, R.; and Onoda, S. 2019. Hyperbolic Disk Embeddings for Directed Acyclic Graphs. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, 6066–6075. PMLR.
- Topping, J.; Giovanni, F. D.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. M. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. In *Proceedings of the 10th International Conference on Learning Representation (ICLR)*.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representation (ICLR)*.
- Wang, H.; Wang, J.; Wang, J.; Zhao, M.; Zhang, W.; Zhang, F.; Xie, X.; and Guo, M. 2018. GraphGAN: Graph Representation Learning With Generative Adversarial Nets. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2508–2515. AAAI Press.
- Wang, S.; Wei, X.; dos Santos, C. N.; Wang, Z.; Nallapati, R.; Arnold, A. O.; Xiang, B.; Yu, P. S.; and Cruz, I. F. 2021. Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion. In *Proceedings of the Web Conference 2021*, 1761–1771. ACM / IW3C2.
- Xiong, B.; Zhu, S.; Nayyeri, M.; Xu, C.; Pan, S.; Zhou, C.; and Staab, S. 2022a. Ultrahyperbolic Knowledge Graph Embeddings. In *Proceedings of the 28th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2130–2139. ACM.
- Xiong, B.; Zhu, S.; Potyka, N.; Pan, S.; Zhou, C.; and Staab, S. 2022b. Pseudo-Riemannian Graph Convolutional Networks. In *Advances in the 35th Conference on Neural Information Processing Systems (NeurIPS)*.
- Yang, H.; Chen, H.; Pan, S.; Li, L.; Yu, P. S.; and Xu, G. 2022. Dual Space Graph Contrastive Learning. In *Proceedings of the ACM Web Conference 2022*, 1238–1247. ACM.
- Yang, M.; Zhou, M.; Pan, L.; and King, I. 2023a. κ HGCN: Tree-likeness Modeling via Continuous and Discrete Curvature Learning. In *Proceedings of the 29th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2965–2977. ACM.
- Yang, M.; Zhou, M.; Ying, R.; Chen, Y.; and King, I. 2023b. Hyperbolic Representation Learning: Revisiting and Advancing. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR.
- Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48, 40–48. JMLR.org.
- Yu, T.; and Sa, C. D. 2023. Random Laplacian Features for Learning with Hyperbolic Space. In *Proceedings of the 11th International Conference on Learning Representation (ICLR)*, 1–23.
- Yu, Z.; and Gao, H. 2022. Molecular Representation Learning via Heterogeneous Motif Graph Neural Networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, 25581–25594. PMLR.
- Zhang, Y.; Wang, X.; Shi, C.; Liu, N.; and Song, G. 2021. Lorentzian Graph Convolutional Networks. In *Proceedings of the Web Conference 2021*, 1249–1261. ACM / IW3C2.
- Zhu, S.; Pan, S.; Zhou, C.; Wu, J.; Cao, Y.; and Wang, B. 2020. Graph Geometry Interaction Learning. In *Advances in the 33rd Conference on Neural Information Processing Systems (NeurIPS)*.