https://doi.org/10.1007/s11704-025-51070-z

RESEARCH ARTICLE

A Riemannian perspective on graph foundation models: curvature as a guiding principle

Li SUN[1]✉, Philip S. YU[2]✉

1. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. Department of Computer Science, University of Illinois Chicago, Chicago 60607-7053, USA

Received July 21, 2025; accepted October 15, 2025

E-mail: lsun@bupt.edu.cn; psyu@uic.edu

© The Author(s) 2025. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

Graphs are ubiquitous, and graph learning has long been a fundamental topic in machine learning. While Graph Neural Networks (GNNs) have achieved remarkable results, they are typically designed for specific tasks or graphs, requiring retraining to adapt the diversity of real-world graph data. Recently, foundation models, such as Large Language Models (LLMs), have driven revolutionary progress in the language domain through universal pretraining. Their success has sparked growing interest in designing Graph Foundation Models (GFMs), a novel family of graph neural networks pre-trained on large-scale, diverse graph data, which are capable of supporting a wide range of downstream tasks on different graphs. Early efforts in the literature often adapt LLMs by transforming graph data into sequential representations. However, unlike word sequences in natural language, graphs are inherently non-Euclidean structures that encapsulate complex intercorrelations among entities. Existing GFMs often trivialize the structural diversity and complexity inherent in graph data. To address this gap, we propose to study graph foundation model from the perspective of Riemannian geometry, and design a novel Curvature-guided Riemannian Graph Foundation Model (CRGFM). To the best of our knowledge, CRGFM is the first GFM to introduce a curvature-based graph description along with geometric standardization. Specifically, to capture structural diversity, the input graph is represented using a mixture of geometric experts. A novel geometric standardization is then introduced via an augmented Lorentz transformation. To model structural complexity, we design a Riemannian graph transformer within a standardized product bundle that disentangles graph structure from node attributes. Finally, we introduce graph prompt learning on the manifold to bridge contrastive learning with downstream tasks. Extensive experiment on a diverse set of real-world graphs demonstrates the superiority of CRGFM.

Keywords

graph neural network; foundation model; riemannian geometry; curvature spaces; prompt learning; self-supervised learning

1 Introduction

Graphs are the neural description of real-world systems with intercorrelated objects, and there exists a diversity of graphs ranging from social media analysis, recommender systems and transportation networks to financial technology, drug discovery, and crystal design [1,2]. Besides the ubiquity, the graphs are complex, presenting a typical non-Euclidean structure compared to the Euclidean ones nested in the language or vision domain [3,4]. In this decade, Graph Neural Networks (GNNs) [5] excel in representation learning on graphs. Although achieving the state-of-the-art performance, the vast majority of GNNs are specialized models, either message-passing neural networks such as graph convolutional nets [6] or the recent graph transformers [7]. Given the graph data, GNNs are typically trained by certain tasks in an end-to-end fashion, and thus a laborious re-training is necessary to maintain the expressiveness for other tasks [8]. Recent years have witnessed the integration between the self-supervised learning and GNNs, e.g., graph contrastive learning [9] and masked graph autoencoders [10]. These methods learn from the graph data themselves and disentangle GNNs from the downstream learning tasks. However, the performance is limited by the gap between self-supervised pre-text and target task [11], and is often unpredictable on the new graphs [12].

Foundation models such as Large Language Models (LLMs) have marked a revolutionary advancement in recent years. In the language domain, LLMs (e.g., GPT4, o1, and Gemini) exhibit the universal expressiveness that a single, pre-trained network is capable of addressing numerous learning tasks on a diversity of datasets [13]. Encouraged by the tremendous success of LLMs, there has recently been a surge of interest in Graph Foundation Models (GFMs) [14]. Conceptually, GFMs refer to a novel family of graph neural networks that are pre-trained on broad graph data at scale and exhibit a deep understanding of underlying structural patterns to support a wide

range of downstream learning tasks (e.g., node classification, clustering, and link prediction) on the diverse real-world graphs. In the literature, existing studies approach GFM mainly from two research directions. The one line leverages LLMs for representation learning, where the graphs are re-casted as sequences readable by LLMs [15], while the other line designs GFM upon GNNs and introduces prompt learning strategies to unify the downstream tasks [16]. So far, GFM is still far from reality, and the significant challenges are discussed below.

Firstly, structural complexity against euclidean backbone networks. Understanding graph structures shows a fundamental importance in learning about graphs. However, the existing GFMs largely ignore the structural complexity in the graph domain, limiting the expressive power. On the one hand, LLM-based methods [17] deconstruct the structural regularity when translating graphs into language-like descriptions, given that the underlying structure of LLM is Euclidean. For example, the order of description affects the performance of representation learning [6], violating a critical topological rule of permutation invariance [18]. Another evidence is given by Section 4 experiment that they often struggle in link prediction and clustering tasks, demonstrating the deficiency in structure understanding. On the other hand, GNN-based methods [5] typically utilize the traditional Euclidean backbone, i.e., the graph convolutional network [6]. In fact, a graph structure aligns with a certain Riemannian manifold, and there exists no isometric mapping between the Riemannian manifold and Euclidean space [19]. Note that these methods cannot be directly transferred to the manifold, given the distinction in formulation and operation owing to the essential difference in geometry. We emphasize that the complex structural patterns[1)] typically go beyond the expressiveness of constant curvature spaces[2)]. In other words, there is an urgent call for an expressive backbone network suitable for the structural complexity underlying the graph domain.

Secondly, structural diversity against the mission of model universality. A key feature of the foundation model is the universality. In the language domain, different datasets present a unified structure of sequence and a shared semantics of word vocabulary [21]. In contrast, different graphs exhibit obvious structural diversity, rendering the construction of GFM even tougher. Specifically, the challenge of structural diversity is how to model the nodes of different graph structures in a unified way. As mentioned above, the language-like graph description is problematic. Recently, analogous to the word vocabulary, [14,22] put forward the notion of structural vocabulary consisting of tree and cycle substructures. However, it is nontrivial to identify the cycles in large graphs [23], and there exists even richer structural semantics in the graph as well. Also, we emphasize that, though Riemannian geometry offers the systematic tools for structural description, the majority of Riemannian graph representation learning is orthogonal to the mission of foundation models [20]. Concretely, they primarily focus on personalized structural matching to geometric counterparts of Riemannian manifolds (e.g., hyperbolic space [3,24], product manifolds [25,26], the κ-stereographic model [27] and pseudo Riemannian manifold [28]), rather than the architecture design for model universality. In short, it still largely remains open to tackle the structural diversity and universality of GFM.

Our solution is a Riemannian perspective on graph foundation models. Grounded on the Riemannian geometry, we rethink the graph foundation model regarding structural complexity and diversity in the graph domain, and propose a novel Curvature-guided Riemannian Graph Foundation Model (CRGFM). The key innovation lies in that we introduce the first curvature-based graph description and its geometric standardization into GFM, to the best of our knowledge. In CRGFM, Riemannian geometry offers the concept of the tangent bundle for the disentanglement between the non-Euclidean graph structure and node attributes, where the Riemannian manifold is employed to depict the graph structure while the section of tangent spaces accommodates node attributes. Specifically, to address the structural diversity, our idea is to describe the input graph by a mixture of geometric experts and then follow up with a novel standardization phase, minimizing the embedding distortion. In particular, the geometric expert is given by the unified formalism in the κ-stereographic model. Input graphs are standardized in the representation space of the product bundle in light of the limited expressiveness of a single geometry, and the geometric standardization is performed through the augmented Lorentz transformation. To address the structural complexity, we propose a Riemannian graph transformer for graph modeling in the standardized product bundle. In particular, the structural encodings are positioned on the manifold by the cross-geometry attention, whose advantage against the traditional attention is shown in the experiment. Meanwhile, the attribute encodings are learned via the aggravation with parallel transport on the manifold. Subsequently, we conduct curvature-based self-supervised learning without task-related annotations, i.e., contrasting positive and negative samples in a shared space. Last but not least, we design a graph prompt learning that a parameterized displacement on the manifold is introduced to perturb the geometric distribution, bridging the gap between the pre-trained model and downstream learning tasks.

The key contributions are three-fold:

- We rethink the graph foundation model regarding structural complexity and diversity in the graph domain, and for the first time connect GFM to geometric standardization in Riemannian geometry for universal graph structural understanding and modeling, to the best of our knowledge.
- We present the novel CRGFM with the representation space of product bundle, where the graphs are described by the mixture of geometric experts, standardized by the augmented Lorentz transformation, encoded by the Riemannian graph transformer, and finally aligned with the target task by prompt learning on the manifold.

[1)] The graph structure is rather complex, negatively curved (hierarchical) in some regions and positively curved (cyclical) in others [20].
[2)] The terminologies of space and manifold are used interchangeably throughout this paper.

- Extensive experiments on real-world graphs demonstrate the superior cross-domain transferability of CRGFM in few-shot learning and zero-shot learning, and we also examine CRGFM effectively supports a wide range of downstream tasks, including node classification, link prediction and node clustering.

The remainder of this paper is organized as follows. We introduce the notations and necessary background in Section 2. Subsequently, Section 3 presents our solution to Geometric Graph Foundation Model (CRGFM). We show the empirical results on a diversity of real-world graphs in Section 4. Section 5 briefly summarizes the related work and specifies their connection to ours. We close with Section 6 of the conclusion and future directions.

2 Preliminaries

In this section, we introduce the necessary background on manifolds, curvature and constant curvature space, and formulate the studied problem of designing a graph foundation model in account of structural complexity and diversity in the graph domain. The important notations are summarized in Table 1.

2.1 Riemannian geometry

Riemannian Geometry offers a systematic construction for structural analysis. The elementary object is called Riemannian manifold, which refers to a smooth manifold $\mathcal{M}^d$ endowed with a Riemannian metric g, where d is the dimensionality. It typically does not obey the usual vector operations but Lie algebra is defined instead. In a Riemannian manifold, each point $x \in \mathcal{M}$ is associated with a tangent space $\mathcal{T}_x\mathcal{M} \in R^d$ around x. A tangent bundle is defined as a smooth manifold $\mathcal{M}$ equipped with a section of disjoint tangent spaces surrounding it $\mathcal{T}\mathcal{M} = \bigsqcup_{x\in\mathcal{M}} \mathcal{T}_x\mathcal{M}$. Riemannian manifold follows the addition group of Lie algebra, so that exponential and logarithmic map are inherited for the projection between the tangent space and the manifold. In particular, the logarithmic map at x does $\log_x(\cdot) : \mathcal{M} \to \mathcal{T}_{\S}\mathcal{M}$, while the exponential map acts inversely. The transform between two tangent spaces is done via parallel transport. The curve of the minimal length connecting two points on the manifold is referred to as a geodesic, which is typically curved rather than straight. The curvature κ_x is a geometric quantity that describes the extent how a surface deviates from being flat at x.

2.2 Constant curvature spaces

A manifold is said to be a Constant Curvature Space (CCS) if and only if its curvature is equal everywhere. Accordingly, there exists three types of constant curvature spaces: hyperbolic space $\mathcal{H}$ with negative curvature, hyperspherical space $\mathcal{S}$ with positive curvature, and "flat" Euclidean space, a special case of zero curvature. Specifically, hyperbolic space excels in modeling tree-like/hierarchical graphs as its intrinsic geometry aligns with the branching construction of such structures. Another evidence is that, for any tree, the embedding distortion[3] is proved to be bounded in low-dimensional hyperbolic spaces, but it is not bounded in any Euclidean space [29]. The flat geometry of Euclidean spaces usually struggles in capturing the irregular structures. Hyperspherical spaces are adept at representing cyclic or symmetric structures, such as rings and cycles. Constant curvature spaces provide significant geometric priors for representation learning, and how to connect them to universality of GFM is still under investigated.

Table 1 Importation notations

Notation	Description
$\mathcal{G}$	Graph
$\mathcal{V}$	Graph nodes set
$\mathcal{E}$	Graph edges set
$\boldsymbol{X}$	Node attributes matrix
$\boldsymbol{A}$	Graph adjacency matrix
$\mathcal{M}$	A smooth manifold
$\mathfrak{g}$	Riemannian metric
$\mathcal{T}_x\mathcal{M}$	The tangent space at $\boldsymbol{x}$
$\mathcal{T}\mathcal{M}$	The tangent bundle surrounding the manifold.
d	Dimension
κ	Curvature
$\ddagger_i$	Structure encoding in tangent space
$\mathcal{H}$	Hyperbolic space
$\mathcal{S}$	Hyperspherical space
$\mathcal{L}$	A unified formalism of Lorentz/Spherical model
$\boldsymbol{o}$	North pole of the model space
$\boldsymbol{p} \in \mathcal{L}$	Node coordinate on the manifold
$\boldsymbol{z} \in \mathcal{T}_{\boldsymbol{p}}\mathcal{L}$	Node encoding in the tangent space
$\phi : \mathcal{L}\times\mathcal{L} \to \mathbb{R}$	A parameterized scalar map
$f(\cdot) : \mathcal{L}^m \to \mathcal{L}^n$	Manifold-reserving linear operation
$[\cdot\|\|\cdot]$	Vector concatenation
$\|\|\cdot\|\|$	$L2$ norm
$\mathrm{Exp}_{\boldsymbol{x}}(\cdot)$	The exponential map
$\mathrm{Log}_{\boldsymbol{x}}(\cdot)$	The logarithmic map
$\mathrm{PT}_{x\to y}(\cdot)$	The parallel transport from $\boldsymbol{x}$ to $\boldsymbol{y}$

2.3 κ-Stereographical model

Constant curvature spaces are instantiated with the model spaces,[4] such as Lorentz model, Poincaré ball model, Poincaré halfplane model and Klein model. In this paper, we construct the formulation

[3] Embedding distortion is defined by the average of $\frac{1}{|\mathcal{V}|^2}\sum_{ij}\left|\frac{d_G(v_i,v_j)}{d(\boldsymbol{x}_i,\boldsymbol{x}_j)} - 1\right|$, where each node $v_i \in \mathcal{V}$ is embedded as $\boldsymbol{x}_i$ in representation space. d_G and d denote the distance in the graph and the space, respectively.

[4] Different model spaces of the same geometry are equivalent to each other in essence.

in the κ-Stereographical Model, which unifies hyperbolic ($\kappa<0$) and hyperspherical ($\kappa>0$) geometries with a single analytical framework in the gyrovector ball. Another advantage is that it converges to the usual Euclidean geometry in the limit of $\kappa=0$. Concretely, a n-dimensional smooth manifold of curvature κ is written as $\mathcal{M}_\kappa^d=\{z\in\mathbb{R}^d|-\kappa\|z\|_2^2<1\}$ equipped with a Riemannian metric $g_z^\kappa=(\lambda_z^\kappa)^2\boldsymbol{I}$, where the conformal factor λ_z^κ is derived as follows:

$$\lambda_z^\kappa=2(1+\kappa\|z\|_2^2)^{-1}. \tag{1}$$

In the hyperspherical regime ($\kappa>0$), it reduces to the classical stereographic sphere model, while recovering the Poincaré ball of radius $R=1/\sqrt{-\kappa}$ for $\kappa<0$. There exists the close-form expression of distance, exponential map, logarithmic map and parallel transport, summarized in Table 2.

2.4 Graphs, GNN, and GFM

An undirected graph is described as $\mathcal{G}=(\mathcal{V},\mathcal{E})$ on the node set $\mathcal{V}$ and edge set $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$. [5)] The nodes are optionally associated with a feature matrix $\boldsymbol{X}\in\mathbb{R}^{N\times F}$, where $|\mathcal{V}|=N$ is the number of nodes, and F denotes the dimension of input features. Note that, non-attributed graphs also exist in the real world. The graph $\mathcal{G}$ can be alternatively expressed in a binary adjacency matrix $\boldsymbol{A}$ and, accordingly, the normalized Laplacian of $\mathcal{G}$ is written as $\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is the diagonal degree matrix. Graph Neural Networks (GNNs) are the dominant solution for learning on graphs. For example, a Message Passing Neural Network (MPNN) recursively aggregates the messages in the neighborhood by K layers, and node representation $\boldsymbol{x}_v$ of the kth layer is updated as follows:

$$\boldsymbol{x}_u^{(k)}=\varphi^{(k)}(\boldsymbol{x}_u^{(k-1)},\mathrm{Agg}(\{\boldsymbol{x}_v^{(k-1)}:(u,v)\in\mathcal{E}\})), \tag{2}$$

for $k=\{1,\ldots,K\}$, where Agg is a permutation invariant aggregation function, and $\varphi^{(k)}$ denotes the update function. Popular MPNNs includes GCN [6], SAGE [5], and GAT [30]. Despite the effectiveness, GNNs are often cast as specialized models, and re-training is typically unavoidable for learning new graphs or new tasks. Encouraged by the success in the language domain, there has recently been a surge of interest in Graph Foundation Models (GFMs), which is a novel family of graph neural networks that are pre-trained on broad graph data at scale and are able to support a wide range of downstream tasks on different graphs. Given a collection of graph $\{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_i,\ldots,\mathcal{G}_M\}$, GFM aims to pre-train a single, universal model $\boldsymbol{\Phi}$ so that the parameter $\boldsymbol{\Theta}$ trained from $\mathcal{G}_i$ can generate informative representations for $\mathcal{G}_j$, $j\in\{1,2,\ldots,M\}$ with slight treatments, e.g., prompt learning. Distinguished from the previous GFMs, we are interested in addressing the graph universality (not limited to textual-attributed graphs) as well as task universality (including node classification, link prediction, and node clustering), where we emphasize the structural complexity and diversity in the graph domain.

3 Methods

Grounded on Riemannian geometry, we propose a novel Curvature-guided Riemannian Graph Foundation Model, referred to as CRGFM, to address the structural complexity and diversity in the graph domain. The overall architecture is sketched in Fig. 1, where a novel standardization phase lies in the heart of CRGFM to achieve universality of foundation models. To the best of our knowledge, we propose the first geometric standardization from the lens of curvature. In a nutshell, for an input graph, it is described by the mixture of geometric experts (Subsection 3.1), standardized by the augmented Lorentz transformation (Subsection 3.2), encoded using Riemannian graph transformer in the latent space (Subsection 3.3), trained with curvature-based self-supervised learning (Subection 3.4)

Table 2 Summary of the operations in constant-curvature space (hyperbolic $\mathbb{H}^d$, spherical $\mathbb{S}^d$, and euclidean space $\mathbb{E}^d$)

Operation	Formalism in $\mathbb{E}^d$	Unified formalism in κ-stereographic model ($\mathbb{H}^d/\mathbb{S}^d$)
Distance metric	$d_{\mathcal{M}}^\kappa(\boldsymbol{x},\boldsymbol{y})=\|\boldsymbol{x}-\boldsymbol{y}\|_2$	$d_{\mathcal{M}}^\kappa(\boldsymbol{x},\boldsymbol{y})=\frac{2}{\sqrt{\|\kappa\|}}\tan_\kappa^{-1}\left(\sqrt{\|\kappa\|}\|-\boldsymbol{x}\oplus_\kappa\boldsymbol{y}\|_2\right)$
Exponential mapping	$\exp_{\boldsymbol{x}}^\kappa(\boldsymbol{v})=\boldsymbol{x}+\boldsymbol{v}$	$\exp_{\boldsymbol{x}}^\kappa(\boldsymbol{v})=\boldsymbol{x}\oplus_\kappa\left(\tan_\kappa\left(\sqrt{\|\kappa\|}\frac{\lambda_{\boldsymbol{x}}^\kappa\|\boldsymbol{v}\|_2}{2}\right)\frac{\boldsymbol{v}}{\sqrt{\|\kappa\|}\|\boldsymbol{v}\|_2}\right)$
Logarithmic mapping	$\log_{\boldsymbol{x}}^\kappa(\boldsymbol{y})=\boldsymbol{x}-\boldsymbol{y}$	$\log_{\boldsymbol{x}}^\kappa(\boldsymbol{y})=\frac{2}{\sqrt{\|\kappa\|}\lambda_{\boldsymbol{x}}^\kappa}\tan_\kappa^{-1}\left(\sqrt{\|\kappa\|}\|-\boldsymbol{x}\oplus_\kappa\boldsymbol{y}\|_2\right)\frac{-\boldsymbol{x}\oplus_\kappa\boldsymbol{y}}{\|-\boldsymbol{x}\oplus_\kappa\boldsymbol{y}\|_2}$
Parallel transport	$PT_{\boldsymbol{x}\to\boldsymbol{y}}^K(\boldsymbol{v})=\boldsymbol{v}-\boldsymbol{x}+\boldsymbol{y}$	$PT_{\boldsymbol{x}\to\boldsymbol{y}}^K(\boldsymbol{v})=-\frac{\lambda_{\boldsymbol{x}}^K}{\lambda_{\boldsymbol{y}}^K}((\boldsymbol{y}\oplus_K-\boldsymbol{x})\oplus_K(\boldsymbol{y}\oplus_K(-\boldsymbol{x}\oplus_K\boldsymbol{v})))$
Addition	$\boldsymbol{x}\oplus_\kappa\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{y}$	$\boldsymbol{x}\oplus_\kappa\boldsymbol{y}=\frac{(1+2\kappa\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}+K\|\boldsymbol{y}\|^2)\boldsymbol{x}+(1-\kappa\|\boldsymbol{x}\|^2)\boldsymbol{y}}{1+2\kappa\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}+\kappa^2\|\boldsymbol{x}\|^2\|\boldsymbol{v}\|^2}$
Scalar-vector multiplication	$r\otimes_\kappa\boldsymbol{x}=r\boldsymbol{x}$	$r\otimes_\kappa\boldsymbol{x}=\exp_{\boldsymbol{0}}^\kappa(r\log_{\boldsymbol{0}}^\kappa(\boldsymbol{x}))$
Matrix-vector multiplication	$\boldsymbol{M}\otimes_\kappa\boldsymbol{x}=\boldsymbol{M}\boldsymbol{x}$	$\boldsymbol{M}\otimes_\kappa\boldsymbol{x}=\exp_{\boldsymbol{0}}^\kappa(\boldsymbol{M}\log_{\boldsymbol{0}}^\kappa(\boldsymbol{x}))$
κ-right multiplication	$X\otimes_k W=XW$	$X\otimes_k W=\exp_0^k(\log_0^k(X)W)$
Applying functions	$f^{\otimes_\kappa}(\boldsymbol{x})=f(\boldsymbol{x})$	$f^{\otimes_\kappa}(\boldsymbol{x})=\exp_{\boldsymbol{0}}^\kappa(f(\log_{\boldsymbol{0}}^\kappa(\boldsymbol{x})))$

[5)] The directed graphs are out of the scope of our study.

or aligned with the target task through prompt learning on the manifold (Subsection 3.5). Note that another advantage of CRGFM is that the geometric structural analysis is not tied to node attributes, and thus broadens the realm of GFM beyond text-attributed graphs. Before elaborating on each component, we specify that the node encoding of CRGFM adopts the disentangled construction of structural encoding and attribute encoding.

3.1 Mixture of geometric experts

In fact, the graph structure is rather complex, hierarchical in some regions and cyclical in others. A single constant curvature space is inadequate to describe the graph structure, given that hyperbolic space is suitable for the hierarchical structures while the cyclical structures call for hyperspherical space in the meantime. Such a challenge motivates us to introduce the Mixture of Geometric Experts (MoGE). Without loss of generality, we introduce K geometric experts and a gating network (a.k.a. router) for node-wise structural description, minimizing the embedding distortion in graph representation.

● Geometric expert in κ-stereographic model

Our geometric expert works with the κ-stereographical model, enjoying its unified formalism in hyperbolic space, hyperspherical space, and Euclidean space. The expert is designed as a Message-Passing Neural Network generalizing Eq. (2) to the manifold. The κ-stereographical MPNN is formulated as follows,

$$\boldsymbol{H}^{(l+1)} = \sigma^{\otimes_\kappa}\left(\hat{\boldsymbol{A}} \boxtimes_\kappa \left(\boldsymbol{H}^{(l)} \otimes_\kappa \boldsymbol{W}^{(l)}\right)\right), \tag{3}$$

where $\boldsymbol{H}$ denotes the representations. $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d\times d}$ is a trainable weight matrix, and $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-1/2}$ is the symmetrically normalized adjacency matrix with added self-loops. The κ-left-matrix-multiplication $\boxtimes_\kappa$ is wiritten as

$$(\boldsymbol{A} \boxtimes_\kappa \boldsymbol{X})_{i\bullet} := \left(\sum_j \boldsymbol{A}_{ij}\right) \otimes_\kappa m_\kappa(\boldsymbol{X}_{1\bullet}, \ldots, \boldsymbol{X}_{n\bullet}; \boldsymbol{A}_{i\bullet}), \tag{4}$$

where $m_\kappa(\cdot)$ denotes the **gyromidpoint** in the κ-stereographical model,

$$m_\kappa(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \boldsymbol{\alpha}) = \frac{1}{2} \otimes_\kappa \left(\sum_{i=1}^n \frac{\alpha_i \lambda_{x_i}^\kappa}{\sum_{j=1}^n \alpha_j (\lambda_{x_j}^\kappa - 1)} \boldsymbol{x}_i\right), \tag{5}$$

with the conformal factor $\lambda_x^\kappa = 2/(1+\kappa\|\boldsymbol{x}\|^2)$. The intuitive understanding of gyromidpoint is given as follows. As illustrated in Fig. 1(b), gyromidpoint is the geometric centroid regarding the distance metric in κ-stereographical model of Riemannian geometry, analogous to the midpoint in Euclidean space. In particular, we further require the condition of $\sum_j \alpha_j(\lambda_{x_j}^\kappa - 1) \neq 0$ for $\kappa > 0$. Notably, the above architecture recovers the Euclidean GCN as $\kappa \to 0$, seamlessly shifting among curved and flat geometries.

● Topological-aware gating network

Given the complexity of graph structure, a node may live in a hierarchical (hyperbolic) dominant substructure where the cyclical

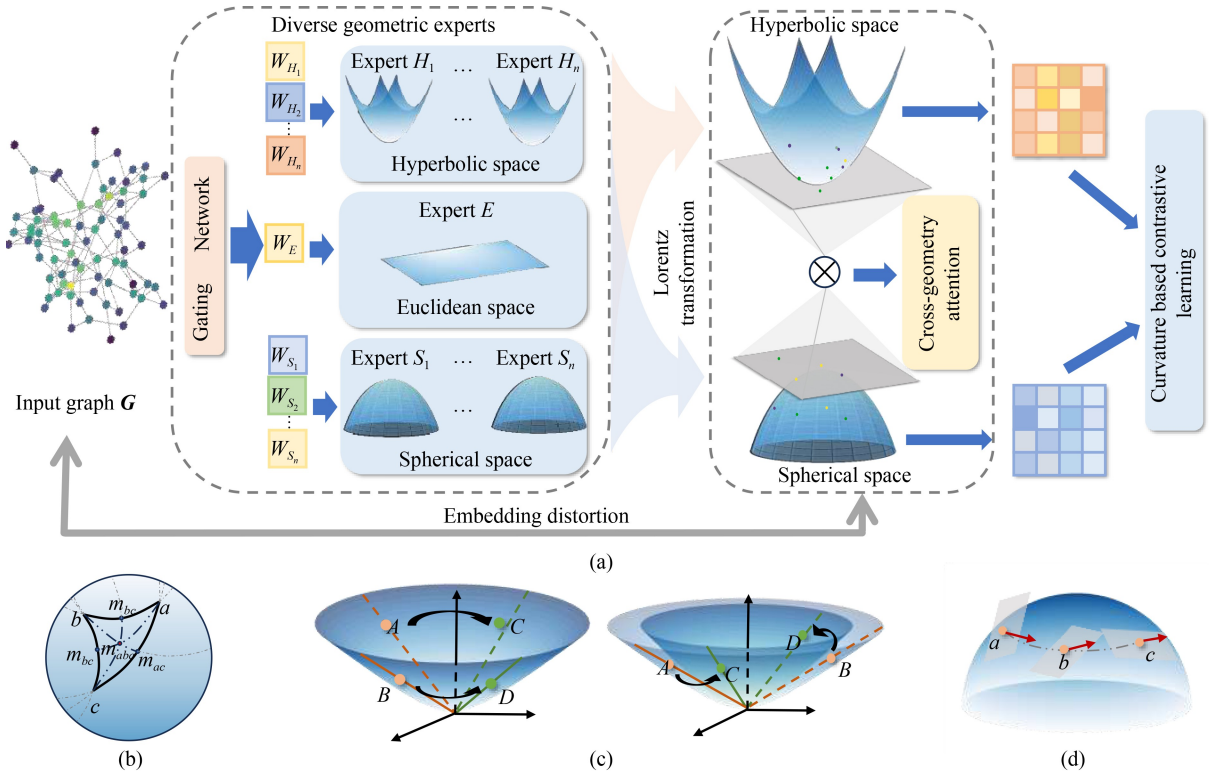


Fig. 1 Overall architecture of the pre-training model in CRGFM. Each graph is described by the mixture of geometric experts, standardized in the product bundle, encoded by the Riemannian graph transformer, and pre-trained with contrastive learning between the hyperbolic and hyperspherical geometries. (a) Overall architecture; (b) gyromidpoint; (c) the lorentz transformation ans augmented lorentz transformation; (d) parallel transport

(hyperspherical) component exists as well. Hence, we introduce K geometric experts of different curvatures, and leverage a gating network that routes the node to the best matching geometric experts. In particular, we adopt the soft assignment that each expert will receive a matching score (weight). In CRGFM, the principle of expert assignment lies in that the underlying geometry of the weighted expert mixture aligns with the local topology of the target node. Accordingly, for a given node i, we first characterize its local topology as a vector representation $\boldsymbol{t}_i$, which is defined as the result of a pooling function over the encodings of the nodes in the subgraph surrounding the target node. Second, we design a network to generate the assignment weight.

$$\boldsymbol{w}_i = \text{softmax}\left(\phi(\boldsymbol{t}_i)\right), \tag{6}$$

where ϕ is a shallow projection such as MLP applied to the subgraph representation. Note that the weight w and encoding $\boldsymbol{h}$ are fed into the geometric standardization, $w \otimes_\kappa \boldsymbol{h}$. Third, we introduce the objective of embedding distortion to measure the agreement.

$$\mathcal{J}_{\text{MoGE}} = \frac{1}{|\mathcal{V}|^2} \sum_{ij} \left| \frac{d_G(v_i, v_j)}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)} - 1 \right|, \tag{7}$$

where each node $v_i \in \mathcal{V}$ is embedded as $\boldsymbol{x}_i$ in representation space. d_G and d denote the distance measure in the graph and the representation space, respectively. Geometrically, each assignment corresponds to a product manifold of the selected experts given the independence of geometric experts. The remaining challenge is to define the distance measure (as stated in the next part), motivating us to propose a standardization phase.

3.2 Geometric standardization

In CRGFM, we propose a novel geometric standardization for the universality under structural diversity. Note that, MoGE describes each node with different product manifolds. The key challenge lies in that these product manifolds, underlying the MoGE, are not comparable. Note that, it is nontrivial to formulate a well-defined measure among a diverisity of product manifolds. Instead, we propose to map different product manifolds into a unified latent space through geometric transformation, i.e., Geometric Standardization. The following subsections detail the construction of unified latent space, the formulation of transformation, and the objective of standardization.

● Representation space of product bundle

Here, we construct the unified latent space for universal graph learning in account of the following two issues. First, given the structural complexity and the importance of the geometric prior, the unified space is preferable to model different structural patterns. It motivates us to adopt the product manifold of multiple geometric measures to address the limited expressive power of a single geometry. Second, in light of the representation disentanglement in our model philosophy, we propose to leverage the concept of the tangent bundle, where structural information and node attributes reside in the different components of the bundle. Consequently, the unified latent space is given as a Product Bundle, which is the Cartesian product of a hyperbolic tangent bundle and a hyperspherical tangent bundle.[6)]

Specifically, the product bundle is written as follows:

$$\mathcal{P}^{d_P} = \left(\mathcal{H}^{d_H}_{\kappa_H} \otimes \mathcal{T}\mathcal{H}^{d_H}_{\kappa_H}\right) \otimes \left(\mathcal{S}^{d_S}_{\kappa_S} \otimes \mathcal{T}\mathcal{S}^{d_S}_{\kappa_S}\right), d_P = 2d_H + 2d_S, \tag{8}$$

where $\otimes$ denotes Cartesian product. $d_{(\cdot)}$ and $\kappa_{(\cdot)}$ stand for the dimensionality and curvature, respectively. There are two factors in this product, i.e., hyperbolic bundle $\mathcal{H}^{d_H}_{\kappa_H} \otimes \mathcal{T}\mathcal{H}^{d_H}_{\kappa_H}$ and hyperspherical bundle $\mathcal{S}^{d_S}_{\kappa_S} \otimes \mathcal{T}\mathcal{S}^{d_S}_{\kappa_S}$, so that we are able to jointly capture hierarchical and cyclical patterns in the latent space.[7)] According to disentanglement, nodes are represented by both structural encoding $\boldsymbol{p}$ in the manifold and attribute encoding $\boldsymbol{z}$ in tangent spaces. Hence, the node representation is constructed as $\boldsymbol{x}_i = [\boldsymbol{p}_i^H \| \boldsymbol{z}_i^H \| \boldsymbol{p}_i^S \| \boldsymbol{z}_i^H] \in \mathcal{P}^{d_P}$, where $\|$ denotes the concatenation operation between vectors, and the four components belong to manifolds and tangent spaces of different geometries $\boldsymbol{p}_i^H \in \mathcal{H}^{d_H}_{\kappa_H}$, $\boldsymbol{z}_i^H \in \mathcal{T}_{\boldsymbol{p}_i^H}\mathcal{H}^{d_H}_{\kappa_H}$, $\boldsymbol{p}_i^S \in \mathcal{S}^{d_S}_{\kappa_S}$, $\boldsymbol{z}_i^S \in \mathcal{T}_{\boldsymbol{p}_i^S}\mathcal{S}^{d_S}_{\kappa_S}$. Accordingly, the Riemannian metric of this product bundle is yielded as $\mathfrak{g}^{\mathcal{P}}_x = \mathfrak{g}^{\kappa_H}_x \oplus \boldsymbol{I}_{d_H+1} \oplus \mathfrak{g}^{\kappa_S}_x \oplus \boldsymbol{I}_{d_H+1}$, where $\boldsymbol{I}_{d_H+1}$ is the (d_H+1)-dimensional identity matrix, and $\oplus$ denotes the matrix direct sum.

● Representation disentanglement and tangent bundle

In this part, we further elaborate on representation disentanglement and product bundle. The structural information and the attribute information are disentangled in the proposed CRGFM, where the manifold is responsible for structural encoding while the tangent spaces for attribute encoding. The final node representation is given as the concatenation of structural encoding and attribute encoding. On the one hand, we model the structural information on the manifold, where the initial input is the eignvalues of Laplacian matrix, which encapsulates the graph structure but is independent to node attributes. The structural information is encoded as the coordinates on the manifold, and we update the coordinates regardless of tangent spaces. The learning on structural knowledge keeps separate to the attribute information. On the other hand, we consider the attribute information on the tangent spaces. Concretely, we conduct neighborhood aggregation for attribute encoding. As attribute encodings live in the tangent spaces, manifold coordinates are needed to bridge the incompatibility between different tangent spaces. Hence, structural information and attribute information are conceptually kept separate yet interact with in the proposed model.

● Augmented lorentz transformation

The standardization is given by mapping the curvature space of each expert to the unified product bundle. Without loss of generality, all the curvature spaces are mapped to the hyperbolic space as well as the hyperspherical space, as shown in Fig. 1. Hence, it calls for a geometric transformation that maps between the manifolds of different curvatures and different dimensionality.

To achieve this, we propose an Augmented Lorentz

6) The hyperbolic tangent bundle refers to the tangent bundle of a hyperbolic space, and is also termed as hyperbolic bundle for short in the following parts.
7) We utilize the subscripts of $\mathcal{H}$ and $\mathcal{S}$ to distinguish the difference in geometry.

Transformation, termed as ALT. We begin by introducing the classic Lorentz transformation in a hyperbolic space. It performs the equivariant transforms such as rotation and boost through a clean, closed form of matrix-vector multiplication, where the multiplier matrix satisfies certain constraints, e.g., orthogonal. However, the constrained optimization is usually infeasible in deep learning. We specify that, expressing linear transformation as a matrix is naturally in Euclidean space, but it is not the case in the manifold endowed with a Riemannian metric. Notably, different from maintaining in the same space as in the Lorentz transformation, we face a much tougher task of simultaneously transforming curvature and dimensionality. In spirit of Lorentz transformation, in ALT, we adopt the clean form of matrix-vector multiplication, but recast the constrained optimization as a network layer. Following the convention of classic formulation, we first elaborate on ALT in the Lorentz/Sphere model of the spacetime formulation. Specifically, it is a smooth manifold $\left\{\begin{bmatrix} x_t \\ \boldsymbol{x}_s \end{bmatrix} \in \mathbb{R}^{d+1} | \langle \boldsymbol{x}, \boldsymbol{x} \rangle_\kappa = \frac{1}{\kappa}, x_t > 0, \boldsymbol{x}_s \in \mathbb{R}^d \right\}$ coupled with the inner-product metric defined as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_\kappa := \mathrm{sgn}(\kappa) x_t y_t + \boldsymbol{x}_s^\top \boldsymbol{y}_s, \tag{9}$$

where x_t and $\boldsymbol{x}_s$ denote the time-dimension and space-dimension, respectively. sgn is the sign function. Let ALT connect the source manifold $\mathcal{M}^{d_1,\kappa_1}$ and target manifold $\mathcal{M}^{d_2,\kappa_2}$, without loss of generality. For any $\boldsymbol{x} \in \mathcal{M}^{d_1,\kappa_1}$, the ALT parameterized by $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$ is formulated as follows:

$$ALT_{d_1,\kappa_1 \to d_2,\kappa_2}(\boldsymbol{W}, \boldsymbol{x}) = \begin{bmatrix} w_0 & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{W} \end{bmatrix} \begin{bmatrix} x_t \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} w_0 x_t \\ \boldsymbol{W} \boldsymbol{x}_s \end{bmatrix}, \tag{10}$$

where $w \in \mathbb{R}$, $w_0 = \sqrt{\frac{|\kappa_1|}{|\kappa_2|} \cdot \frac{1 - \kappa_2 \ell(\boldsymbol{W}, \boldsymbol{x}_s)}{1 - \kappa_1 \langle \boldsymbol{x}_s, \boldsymbol{x}_s \rangle}}$, and $\ell(\boldsymbol{W}, \boldsymbol{x}_s) = \|\boldsymbol{W} \boldsymbol{x}_s\|^2$. Then, we rewrite ALT in the κ-stereographical model[8], which is done via stereographical projection. In particular, it is a diffeomorphism connecting the different model spaces

$$\boldsymbol{x} = \Pi(\boldsymbol{x}') = \frac{1}{1 + \sqrt{|\kappa|} x_t'} \boldsymbol{x}_s', \quad \boldsymbol{x}' = \Pi^{-1}(\boldsymbol{x}) = \left(\lambda_{\boldsymbol{x}}^\kappa \boldsymbol{x}, \frac{1}{\sqrt{|\kappa|}} (\lambda_{\boldsymbol{x}}^\kappa - 1) \right), \tag{11}$$

where $\boldsymbol{x}'$ is a point on the Lorentz/Sphere model, and the image $\boldsymbol{x}$ is the corresponding point in the gyrovector ball. Finally, in $\kappa-$stereographic model, ALT takes the form of

$$\Pi\left(ALT_{d_1,\kappa_1 \to d_2,\kappa_2}(\boldsymbol{W}, \Pi^{-1}(\boldsymbol{x})) \right). \tag{12}$$

Here, we review the concept of Lorentz transformation to develop the high-level understanding of ALT. Lorentz transformation can be regarded as the linear transformation in a given hyperbolic space, steaming from the Einstein's Special Theory of Relativity. The proposed ALT generalizes the Lorentz transformation and allows for the linear transformation between any two κ-stereographical model. In other words, the realm of Lorentz transformation is extended from the hyperbolic space to a wider family of constant curvature spaces. Accordingly, we are able to implement the regularization of embedding distortion in the standardized product bundle with augmented Lorentz transformation. In particular, the distance is given by the product of hyperbolic and hyperspherical space, where we utilize the gyromidpoint with the weights of geometric experts in each CCS.

3.3 Riemannian graph transformer

In CRGFM, we design a Riemannian graph transformer on the standardized product bundle to learn expressive encodings. We notice that, recent advance extends the vanilla transformer to the hyperbolic space [31]. Key differences are dual: on the one hand, we are devoted to deriving a unified formalism for constant curvature spaces (the formalism is preferable to work with both hyperbolic space and hyperspherical sapce, and recovers to its Euclidean counterpart in the limit of curvature at zero); on the other hand, our Riemannian transformer are designed to instantiate attention mechanism over the tangent spaces, which has rarely been touched in the literature.

● Structural encoding with cross-geometry attention

The structural encoding is the node's location in the manifold, depicting its structural regularity in the graph. Towards this end, we adopt the global attention mechanism in order to capture the global structure context, so that the universal mechanism to represent structural regularity is injected into the model parameters. In other words, given an arbitrary graph structure, the proposed manifold attention is responsible to learn the node distribution from the corresponding geometric perspective.

Specifically, we generalize the vanilla global attention to constant curvature spaces. Note that, we derive the initial structural encoding with the graph Laplacian, which is manifold-valued by default. We elaborate on the proposed attention mechanism with the hyperbolic factor of the product bundle.[9] In our design, the key, query and value are given by the linearity on the manifold as follows:

$$\boldsymbol{k}_i = \boldsymbol{W}_K \otimes_\kappa \boldsymbol{p}_i^{\mathcal{H}}, \quad \boldsymbol{q}_i = \boldsymbol{W}_Q \otimes_\kappa \boldsymbol{p}_i^{\mathcal{S}}, \quad \boldsymbol{v}_i = \boldsymbol{W}_V \otimes_\kappa \boldsymbol{v}_i^{\mathcal{H}}, \tag{13}$$

where $\otimes_\kappa$ denotes the matrix-vector multiplication in κ-stereographical model. Note that, the query is given by the counterpart geometry in the proposed cross-geometry attention. The attentional weight is defined by the softmax function as follows:

$$\alpha_{ij} = \frac{\exp(\phi([\boldsymbol{q}_i \| \boldsymbol{k}_j]))}{\sum_{j' \in \mathcal{V}} \exp(\phi([\boldsymbol{q}_i \| \boldsymbol{k}_{j'}]))}, \tag{14}$$

where $\|$ denotes the vector concatenation, and ϕ is a scalar scoring function. The cross-geometry attention conducts the global attention over the node set $\mathcal{V}$. Thus, the structural encoding is updated by the weighted aggregation in the gyrovector ball, which is written as gyromidpoint $m_\kappa(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n; \boldsymbol{\alpha})$ in Eq. (5).

Accordingly, the manifold attention learns structural patterns with the geometric prior. We argue that the cross-attention fuses the different geometries as a whole. Also, the ablation study shows its superior expressiveness compared to single-geometry counterpart.

[8] Though the model spaces present different formal construction, they are the same in essence.
[9] The attention in the hyperspherical factor has the same construction with the swapped superscript.

● Attribute encoding through parallel transport

The structural encodings live in the same manifold, either hyperbolic or hyperspherical. However, attribute encodings reside in different tangent spaces, which poses a fundamental challenge that attribute encodings are incompatible.

In CRGFM, we leverage the geometric operation of parallel transport to address the incompatibility among different tangent spaces. Specifically, given two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the manifold, the parallel transport w.r.t. the Levi-Civita connection $PT_{x\to y}$ is a linear isometry that maps a vector in one tangent space $\boldsymbol{v}\in\mathcal{T}_x\mathcal{L}$ to another tangent space $\mathcal{T}_y\mathcal{L}$ along the geodesic connecting $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{L}$. That is, the parallel transport can be given by the movement along the geodesic, whose unit speed is written $\gamma_{x,y}(t)=\boldsymbol{x}\cos_\kappa(t)+\frac{1}{\sqrt{|\kappa|}}\sin_\kappa(t)\boldsymbol{y}$ from $\boldsymbol{x}$ to $\boldsymbol{y}$ is given by, for $t\in[0,1]$. Hence, the general form is expressed with the logarithmic map as follows:

$$PT_{\boldsymbol{p}_i\to\boldsymbol{p}_j}(\boldsymbol{z})=\boldsymbol{z}-\frac{\langle\log^\kappa_{\boldsymbol{p}_i}(\boldsymbol{p}_j),\boldsymbol{z}\rangle_{\boldsymbol{p}_i}}{d(\boldsymbol{p}_i,\boldsymbol{p}_j)}\left(\log^\kappa_{\boldsymbol{p}_i}(\boldsymbol{p}_j)+\log^\kappa_{\boldsymbol{p}_j}(\boldsymbol{p}_i)\right),\tag{15}$$

where the inner product at point $\boldsymbol{x}$ is defined as $\langle\boldsymbol{a},\boldsymbol{b}\rangle_{\boldsymbol{x}}=\boldsymbol{a}^\top\mathfrak{g}_{\boldsymbol{x}}\boldsymbol{b}$, and $\mathfrak{g}_{\boldsymbol{x}}$ denotes the Riemannian metric of $\mathcal{L}$ at $\boldsymbol{x}$. In κ-stereographic model, parallel transport is reduced as the clean expression as follows:

$$PT^\kappa_{\boldsymbol{p}_i\to\boldsymbol{p}_j}(\boldsymbol{z})=-\frac{\lambda^\kappa_{\boldsymbol{p}_i}}{\lambda^\kappa_{\boldsymbol{z}}}\left((\boldsymbol{p}_j\oplus_\kappa-\boldsymbol{p}_i)\oplus_\kappa(\boldsymbol{p}_j\oplus_\kappa(-\boldsymbol{p}_i\oplus_\kappa\boldsymbol{z}))\right),\tag{16}$$

where $\oplus_\kappa$ is the addition given in Table 2.[10] More intuitively, the operation of parallel transport is analogous to the translation in Euclidean space. Parallel transport can be understood as the vector movement along the geodesic on the Riemannian manifold, where the geodesic is the shortest curve connecting two points on the manifold. Thereafter, we are able to conduct local attention among attribute encodings in different tangent spaces. Specifically, for a target node, we bring the attribute encodings of the neighbor nodes to its tangent space with parallel transport. In light of the Euclidean geometry of the manipulation tangent space, we perform the local attention via a graph attention layer of [30] on the local subgraph surrounding the target node.

In the standardized product bundle, we establish a universal mechanism that learns the structural regularity in the manifolds while extracting node attributes in the tangent spaces.

3.4 Geometric self-supervised learning

This part elaborates on how to pre-train CRGFM, and the overall procedure is summarized in Algorithm 1. As the training with task-specific annotations tends to result in inductive bias, the pre-training of foundation models generally follows the self-supervised fashion. Among self-supervised learning methods, contrastive learning has emerged as a successful paradigm where data augmentation is required to learn from the similarity among data themselves. Unlike the image domain, where augmented samples are readily generated by operations such as cropping or rotation, it is inherently challenging to generate meaningful augmented graphs.

Instead of explicit graph augmentation, in CRGFM, we explore the diversity of geometries in the product bundle and enable augmentation-free graph contrastive learning. To be specific, the product bundle itself consists of hyperbolic and hyperspherical geometry, and thus offers different geometric views for graph contrastive learning. Next, we study the similarity measure for the geometric contrast. The key challenge lies in the fact that different geometries are not compatible, and the case becomes more complicated when the tangent bundle is taken into account. To resolve the incompatibility issue, we consider the shared tangent space anchored at the north pole of the model space and, accordingly, the attribute at its structural position is parallel transported to the tangent space at the north pole. To improve efficiency, we only perform contrastive learning among attribute encodings, since the parameters related to the manifolds are encapsulated in parallel transport. Finally, the contrastive learning objective is derived as follows:

$$\mathcal{J}_{\text{SSL}}=\mathcal{J}(H,S)+\mathcal{J}(S,H),\tag{17}$$

$$\mathcal{J}(H,S)=-\sum_{i=1}^{N}\log\frac{\exp(\langle PT_{\boldsymbol{p}_i^H\to\boldsymbol{o}}(\boldsymbol{z}_i^H),PT_{\boldsymbol{p}_i^S\to\boldsymbol{o}}(\boldsymbol{z}_i^S)\rangle)}{\sum_{j=1}^{N}\exp(\langle PT_{\boldsymbol{p}_i^H\to\boldsymbol{o}}(\boldsymbol{z}_i^H),PT_{\boldsymbol{p}_j^S\to\boldsymbol{o}}(\boldsymbol{z}_j^S)\rangle)},\tag{18}$$

where N is the number of nodes. Note that, the pre-training process is of fundamental importance to CRGFM, which learns the universal structural knowledge in the graph domain.

Algorithm 1 Pre-training CRGFM

Input: Pre-training graphs, Number of geometric experts, and Hyperparameters.
Output: Model parameters of `CRGFM`.

```
Normalize node structural inputs by the K largest eigenvalues of the Laplacian matrix;
Initialize model parameters;
while model not converged do
    Compute expert assignment weights with topological-aware gating network;
    Generate structural description with the Mixture of Geometric Experts (MoGE);
    Conduct geometric standardization with Augmented Lorentz Transformation (ALT);
    for each geometry in the standardized bundle do
        Update structural encoding by cross-geometry attention in Eq. (14);
        Update attribute encoding by local attention with the parallel transport in Eq. (16);
    Generate the hyperbolic and hyperspherical views for contrastive learning;
    Compute the embedding distortion loss in Eq. (7) and geometric contrastive loss in Eq. (17);
    Update model parameters via gradient descent.
```

3.5 Riemannian prompt learning on graphs

Prompt learning is typically adopted to bridge the gap between model pre-training and the target learning task. In the literature, a series of

[10] The Möbius addition in the gyrovector ball.

works have designed advanced prompt learning to improve the task universality of graph neural networks [16]. However, all of them are formulated in the traditional Euclidean space, and cannot be directly adjusted to Riemannian manifolds.

To fill this gap, we propose the first Riemannian Prompt Learning for graph foundation model, to the best of our knowledge, and the prompt learning procedure is summarized in Algorithm 2. Its intuitive understanding is given as follows: we perturb the encoding distribution on the manifold with parameterized displacement to align with the target learning task. To be specific, we first froze the pre-trained parameters Φ, initializing a task-specific prompt matrix $\boldsymbol{Q}_t$. For each node v_i within every factor bundle of the product bundle, first project the prompt vector $\boldsymbol{q}_i$ onto the manifold $\boldsymbol{p}_i$ to a new position $\boldsymbol{p}_i'$ via the exponential map:

$$\exp_{\boldsymbol{p}_i}^{\kappa}(\boldsymbol{q}_i) = \boldsymbol{p}_i \oplus_{\kappa} \left(\tan_{\kappa} \left(\sqrt{|\kappa|} \frac{\lambda_{\boldsymbol{p}_i}^{\kappa} \|\boldsymbol{q}_i\|_2}{2} \right) \frac{\boldsymbol{q}_i}{\sqrt{|\kappa|}\|\boldsymbol{q}_i\|_2} \right). \tag{19}$$

Subsequently, parallel transport transfers the attribute encoding z_i from its original position $\boldsymbol{p}_i$ to $\boldsymbol{p}_i'$, ensuring metric invariance during feature translation. Finally, optimize the prompt matrix based on the task-specific loss function to generate node representations that integrate universal structural knowledge with task-adaptive features.

The proposed CRGFM follows the "pre-training and prompt learning" paradigm, where Riemannian prompt learning seamless integrates task-specific information with the pre-training in Riemannian manifolds. Note that prompt vectors for node representations are optimized separately for each downstream task, bridging the gap between the pre-trained model and downstream task.

- Computational complexity. We conduct complexity analysis for CRGFM in both pre-training and prompting. In model pre-training, performing MoGE of K geometric experts costs $O(K|\mathcal{E}|)$ as each expert undergoes the message-passing of $O(|\mathcal{E}|)$. The complexity of Riemannian transformer is yielded as $O(2|\mathcal{V}|^2)$ owing to global attention. It is in the same rank as the traditional transformer but is endowed with rich geometries. The proposed contrastive learning costs $O(|\mathcal{V}|^2)$ while taking advantage of augmentation free. In the prompt learning, each node is attached with a parameterized displacement, and the complexity is given as $O(|\mathcal{V}|)$ for the target tasks of node classification and link prediction.

Algorithm 2 Riemannian prompt learning

Input: Graph G, pretrained parameters Φ, prompt vector $\boldsymbol{q}$
Output: Task-specific node representation.

```
Initialize prompt matrix Q_t ∈ ℝ^{n×d};
Froze pretrained parameters Φ;
for each node v_i ∈ G in each factor bundle do
    Project q_i onto manifold at p_i to get new position p_i' using exponential map;
    Parallel transport attribute encoding z_i at original position p_i to p_i';
Train the prompt matrix Q_t with the task-specific loss;
return Task-specific node representation.
```

- Connection to existing GFMs. Previous efforts can be roughly divided into two groups: LLM-based methods and GNN-based methods. The proposed CRGFM belongs to the latter, and CRGFM distinguishes itself by the rich geometries in structural understanding. In the literature, RiemannGFM [22] is the first Riemannian foundation model in the graph domain, to the best of our knowledge. The proposed model and RiemannGFM work with different model space. More importantly, RiemannGFM introduces the notion of structural vocabulary and explicit samples shared substructures. On the contrary, we model the structural complexity through MoGE and put forward a novel geometric standardization phase for universality.

4 Experiments

We do experiments on a diversity of real-world graphs, aiming to answer the following research questions:

- RQ1: How does the **CRGFM** perform on cross-domain adaption?
- RQ2: How effective is **CRGFM** in few-shot learning?
- RQ3: How does the pre-training dataset impact **CRGFM** performance?

4.1 Experimental setups

4.1.1 Datasets

We conduct extensive experiments on benchmark graph datasets. Without loss of generality, we choose two categories of datasets: text-attributed graphs (Citeseer and Pubmed [32], Amazon-photo [8]) and a non-attributed graph (Airports [32]) for model evaluation. Citeseer and Pubmed are citation networks, where nodes represent documents and edges represent citation links. Amazon-photo is a segment of the Amazon co-purchase graph, where nodes represent goods and edges represent that two goods are frequently bought together. Airport is a commercial air transportation network within the United States, which is a non-attributed graph. For the pre-training datasets, we choose three widely used graph datasets, Flickr, Acomp, and WikiCS. The details are summarized in Table 3.

4.1.2 Baseline

We compare the proposed CRGFM with the following strong baselines categorized into three groups: vanilla GNNs (i.e., GCN and SAGE), graph self-supervised learning methods (i.e., DGI and GraphMAE2) and graph foundation models (i.e., GCOPE, OFA,

Table 3 Statistics of datasets

Dataset	#(Nodes)	#(Edges)	Feature dimension
CiteSeer	3,327	9,104	3,703
Pubmed	19,717	44,338	500
Amazon-photo	7,650	238,162	745
Airports	1,190	13,599	0
ogbn-arxiv	169,343	1,166,243	128
Physics	34,493	495,924	8,415
DBLP	17,716	105,734	1,639

RiemannGFM, OpenGraph, and LLaGA). We briefly introduce the baselines as follows.

- GCN [6] leverages spectral graph convolution to learn node representations by iteratively aggregating and transforming normalized neighbor features to capture graph local structure.
- GraphSAGE [5] samples a fixed number of neighbors for each node and applies a learnable aggregator to combine their features, enabling inductive and scalable node embedding.
- DGI [33] introduces a self-supervised paradigm by maximizing the mutual information between the local and global views.
- GraphMAE2 [10] conducts self-supervised learning in the reconstruction of masked node features with masked autoencoders.
- GCOPE [34] is a graph pre-training framework designed to enhance the efficacy of downstream tasks by harnessing collective insights from multiple source domains.
- OFA [35] describes all nodes and edges with natural language to feed into LLMs, and subsequently utilizes graph prompting that appends prompting substructures to the input graph.
- RiemannGFM [22] models graph learning from the perspective of Riemannian geometry, developing a graph foundation model that leverages geometric properties to capture complex non-Euclidean structures in graphs.
- OpenGraph [36] is trained on diverse datasets with a unified graph tokenizer, scalable graph transformer, and LLM-enhanced data augmentation to comprehend the nuances of diverse graphs.

4.1.3 Implementation notes and evaluation protocol

In CRGFM, the curvatures of hyperbolic and hyperspherical factors in the standardized bundle are fixed as -1 and $+1$, respectively. We utilize 11 geometric experts by default, where a geometric expert of zero curvature is introduced for the flat geometry and the curvatures of other experts are learnable. On input initialization, the structural input is given by the K largest eigenvalues of the Laplacian matrix, where K is predefined number. The model is optimized by Adam with a dropout rate of 0.3. All baseline models are adopted from official repositories. Popular metrics of classification accuracy (ACC) and weighted F1-score (F1) are employed in node classification, while AUC and Average Precision (AP) are utilized in link prediction. Node clustering results are visualized for intuitive description. On the few-shot learning, we follow the evaluation protocol commonly adopted in recent graph foundation models. Specifically, for node classification, we randomly sample 1-shot or 5-shot labeled nodes per class, and fine-tune the pre-trained model using a supervised cross-entropy loss for 100 epochs. Both Accuracy and F1 are reported based on an average over random splits. For link prediction, we adopt the 1-shot or 5-shot setting by randomly revealing a limited number of positive links, combined with an equal number of negative samples. A decoder is trained on the resulting few-shot link pairs, and performance is evaluated using AUC and AP. In all cases, the fine-tuning is performed using early stopping based on validation performance. Each case undergoes 10 independent runs, and we report the mean value with standard deviations.

4.2 Performance evaluation

We evaluate the performance of GFM with knowledge transfer among different graphs and few-shot learning. Then, we investigate the impact of pre-training datasets to show the universality of the proposed model.

4.2.1 Cross-domain transfer learning

Tables 4 and 5 summarize the results of the node classification and link prediction tasks in the cross-domain transfer setting. Note that, OFA cannot work without node attributes. We pre-train the proposed CRGFM on four large-scale datasets. For the specialized models of GCN and GraphSAGE, they are trained directly on the target datasets. According to the experimental results, we summarize the key findings as follows. First, the CRGFM achieves outstanding performance on both node classification and link prediction tasks, validating the effectiveness of our model for cross-domain transfer. We will further analyze the specific contributions of these components in the ablation studies. Second, models relying on LLMs

Table 4 Cross-domain transfer learning on CiteSeer, Pubmed, Amazon-photo, and Airport datasets. ACC (%) and F1 (%) of node classification with standard deviations. The best method in each column is bold, and the runner-up is underlined

Metric	Dataset	Vanilla GNNs		Graph SSL		Graph foundation models					
		GCN	GraphSAGE	DGI	GraphMAE2	GCOPE	OFA	OpenGraph	LLaGA	RiemannianGFM	**CRGFM**
Citeseer	ACC	71.39 ± 0.35	67.13 ± 0.28	72.23 ± 0.31	**73.18 ± 0.25**	65.57 ± 0.29	59.36 ± 0.27	58.95 ± 0.30	59.71 ± 0.26	66.37 ± 0.73	67.48 ± 0.63
	F1	70.43 ± 0.42	67.20 ± 0.35	70.32 ± 0.38	**73.54 ± 0.32**	65.67 ± 0.36	59.36±0.34	58.95 ± 0.37	59.79 ± 0.33	66.46 ± 0.67	68.12 ± 0.64
Pubmed	ACC	75.90 ± 0.21	77.73 ± 0.19	76.13 ± 0.20	**82.27 ± 0.17**	74.34 ± 0.18	75.42 ± 0.16	57.49 ± 0.17	70.88 ± 0.15	76.27 ± 0.38	77.86 ± 0.74
	F1	74.33 ± 0.33	75.69 ± 0.31	75.17 ± 0.32	**79.81 ± 0.29**	73.38 ± 0.30	72.64 ± 0.28	53.32 ± 0.29	63.89 ± 0.27	75.83 ± 0.39	76.58 ± 0.38
Photo	ACC	85.67 ± 0.18	87.92 ± 0.16	86.40 ± 0.17	88.04 ± 0.14	87.61 ± 0.15	88.87 ± 0.13	88.55 ± 0.14	84.12 ± 0.12	89.95 ± 0.89	**90.73 ± 0.48**
	F1	85.56 ± 0.27	87.81 ± 0.25	89.29 ± 0.26	88.93 ± 0.23	86.50 ± 0.24	88.76 ± 0.22	85.44 ± 0.23	74.01 ± 0.21	89.68 ± 0.47	**90.42 ± 0.69**
Airport	ACC	49.30 ± 0.25	49.95 ± 0.23	50.63 ± 0.24	52.61 ± 0.21	39.95 ± 0.22	–	41.46 ± 0.21	36.56 ± 0.19	55.23 ± 0.91	**56.78 ± 0.38**
	F1	48.26 ± 0.31	48.35 ± 0.29	48.79 ± 0.30	48.97 ± 0.27	35.67 ± 0.28	–	37.21 ± 0.27	38.72 ± 0.25	53.17 ± 0.50	**55.82 ± 0.49**

Table 5 Cross-domain transfer learning on CiteSeer, Pubmed, Amazon-photo, and Airport datasets. AUC (%) and AP (%) of link prediction with standard deviations. The best method in each column is bolded, and the runner-up is underlined

Metric	Dataset	Vanilla GNNs		Graph SSL		Graph foundation models					
		GCN	GraphSAGE	DGI	GraphMAE2	GCOPE	OFA	OpenGraph	LLaGA	RiemannianGFM	**CRGFM**
Citeseer	AUC	90.31 ± 0.35	89.34 ± 0.28	96.24 ± 0.31	93.78 ± 0.25	88.63 ± 0.29	83.38 ± 0.27	75.92 ± 0.30	86.45 ± 0.26	99.36 ± 0.14	**99.41 ± 0.29**
	AP	91.85 ± 0.42	88.68 ± 0.35	94.93 ± 0.38	89.28 ± 0.32	83.59 ± 0.36	82.76 ± 0.34	77.43 ± 0.37	83.50 ± 0.33	98.22 ± 0.64	**99.01 ± 0.28**
Pubmed	AUC	92.81 ± 0.21	88.15 ± 0.19	88.49 ± 0.20	89.39 ± 0.17	90.36 ± 0.18	92.10 ± 0.16	70.38 ± 0.17	84.35 ± 0.15	94.18 ± 0.33	**95.47 ± 0.95**
	AP	90.48 ± 0.33	86.69 ± 0.31	87.03 ± 0.32	85.81 ± 0.29	86.30 ± 0.30	91.64 ± 0.28	71.33 ± 0.29	79.29 ± 0.27	91.39 ± 0.32	**91.58 ± 0.63**
Photo	AUC	88.91 ± 0.18	89.72 ± 0.16	94.39 ± 0.17	96.26 ± 0.14	89.75 ± 0.15	94.52 ± 0.13	84.55 ± 0.14	89.12 ± 0.12	95.47 ± 0.39	**96.38 ± 0.91**
	AP	86.27 ± 0.27	87.63 ± 0.25	92.39 ± 0.26	94.39 ± 0.23	86.50 ± 0.24	92.76 ± 0.22	82.44 ± 0.23	87.01 ± 0.21	93.29 ± 0.45	**94.29 ± 0.48**
Airport	AUC	92.30 ± 0.25	92.75 ± 0.23	92.03 ± 0.24	88.67 ± 0.21	86.24 ± 0.22	–	85.18 ± 0.21	75.75 ± 0.19	93.65 ± 0.26	**94.78 ± 0.39**
	AP	93.84 ± 0.31	91.70 ± 0.29	90.38 ± 0.30	90.82 ± 0.27	83.39 ± 0.28	–	84.33 ± 0.27	70.90 ± 0.25	96.19 ± 0.15	**96.75 ± 0.24**

or textual attributes (e.g., OFA, LLaGA) exhibit significant performance degradation on datasets without text attributes (Airport). However, the cross-domain training method GCOPE, which is a text-free cross-domain pre-training method, lags notably behind CRGFM. This is because GCOPE only considers feature alignment during cross-domain transfer learning while neglecting graph structure, further showcasing the critical importance of structural features.

4.2.2 Few-shot learning

Table 6 reports the results of node classification under 1-shot and 5-shot settings, where the GFMs are pre-trained on large-scale datasets and then fine-tuned under few-shot settings for experimental validation. First, the proposed CRGFM outperforms all the baselines across the four datasets, demonstrating its knowledge transfer and generalization capability. In particular, CRGFM exhibited significant performance advantages over existing GFMs on datasets with non-textual features (e.g., Airport). Second, the methods relying on LLM fail to achieve performance improvement and even show negative transfer. It demonstrates the advantage of capturing structure knowledge in the cross-domain transfer, and motivates the design of our model.

4.2.3 Performance of different pre-training datasets

Figure 2 illustrates the impact of different pre-training datasets on the performance of the CRGFM model. Under the experimental setup detailed in Subsection 4.1, we employ Flickr, Acomp, and WikiCS as pre-training datasets. To evaluate the influence of the pre-training dataset composition on model performance, we define three distinct pre-training configurations: 1) All datasets except the target: Pre-training on all available datasets, excluding the target dataset. 2) Only the target dataset: Pre-training solely on the target dataset. 3) All datasets: Pre-training on the full set of available datasets. For these three scenarios, we selected the following baseline models respectively: GCOPE, GCN, DGI, and GraphMAE2. It can be observed that CRGFM consistently outperforms the other baseline models, regardless of the specific pre-training dataset configuration used. This robust performance across diverse data sources demonstrates that CRGFM effectively captures the universal structural patterns underlying the graph domain.

4.3 Ablation study

In this section, we conduct an ablation study to evaluate each

Table 6 1-shot and 5-shot performance on CiteSeer, Pubmed, GitHub, and Airport datasets. The best method in each column is bold, and the runner-up is underlined

Metric	Citeseer		Pubmed		Amazon-photo		Airport	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
DGI	37.68 ± 4.12	46.57 ± 3.76	39.38 ± 2.87	51.76 ± 2.79	58.29 ± 3.96	67.82 ± 2.85	30.60 ± 5.19	37.22 ± 3.01
GraphMAE2	34.04 ± 2.77	48.74 ± 1.67	39.54 ± 4.87	53.80 ± 2.57	59.39 ± 4.10	65.29 ± 3.70	28.37 ± 6.13	38.19 ± 2.05
OFA	38.21 ± 4.85	32.48 ± 2.75	40.69 ± 2.97	37.30 ± 1.06	49.23 ± 3.60	60.32 ± 1.34	–	–
GCOPE	35.74 ± 6.38	44.89 ± 4.14	37.81 ± 4.19	45.86 ± 2.70	57.88 ± 1.67	74.67 ± 1.07	26.59 ± 7.10	36.50 ± 3.62
OpenGraph	21.95 ± 4.86	29.04 ± 3.87	43.97 ± 2.88	37.89 ± 4.01	45.76 ± 4.88	59.34 ± 2.58	31.77 ± 6.98	33.01 ± 2.89
RiemannianGFM	38.38 ± 3.57	53.20 ± 2.58	45.30 ± 4.58	66.30 ± 2.58	75.39 ± 1.02	82.58 ± 0.94	32.78 ± 2.05	38.56 ± 1.56
LLaGA	18.49 ± 4.39	25.40 ± 3.98	35.60 ± 5.99	32.50 ± 4.56	39.59 ± 6.49	42.78 ± 4.78	24.69 ± 7.39	31.59 ± 4.99
CRGFM	**39.40 ± 4.69**	**55.78 ± 2.01**	**46.89 ± 2.74**	**68.11 ± 2.86**	**76.20 ± 1.78**	**83.75 ± 2.69**	**33.78 ± 3.85**	**39.77 ± 2.89**

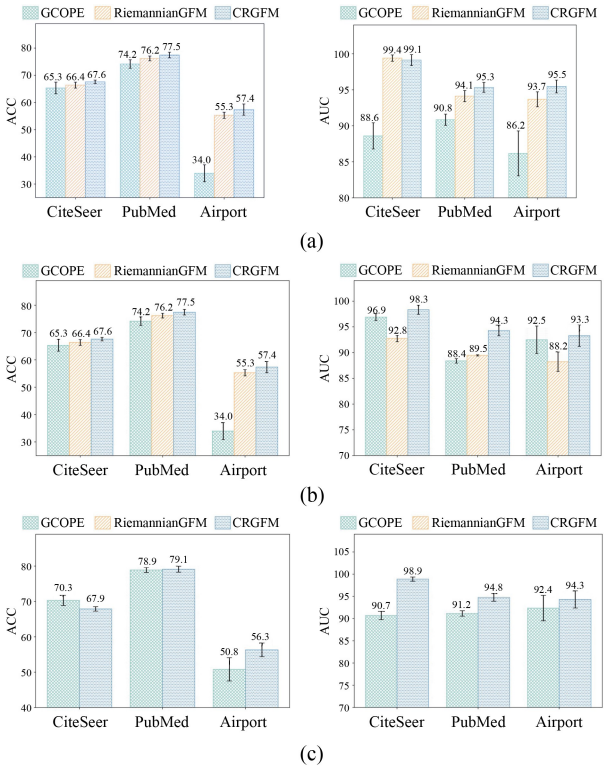


Fig. 2 The impact of different pre-training datasets. (a) Pre-training on all datasets; (b) pre-training on all datasets excluding target dataset; (c) pre-training on target dataset

proposed component of CRGFM. To achieve this, we introduce several variants and the experimental results are summarized in Table 7.

4.3.1 Mixture of geometric experts vs. learnable κ-GCN

To investigate the importance of curvature-based structural description, we compare our model with a variant termed `w/oMoGE`, which replaces the Mixture of Geometric Experts (MoGE) with a single κ-GCN whose curvature is jointly learned during training. In the original MoGE framework, each region of the graph can be matched to a set of geometric experts —hyperbolic, Euclidean, or spherical —based on its structural properties, such as hierarchy, cycles, or symmetries. In contrast, this variant removes the region-wise assignment of geometric priors and instead enforces a shared curvature scalar across all nodes. As quantitatively shown in Table 7, Learnable κ-GCN consistently underperforms the full MoGE-enhanced model across all benchmarks. The performance drop is particularly significant on datasets such as Pubmed and Airport, where graphs exhibit mixed structural regimes—hierarchical tree-like backbones and local ring structures. This highlights that a globally learned curvature is insufficient for capturing fine-grained structural variations, and confirms that MoGE's geometry-specific experts are critical for modeling graphs with multi-scale or mixed-topology features.

4.3.2 Augmented lorentz transformation vs. MLP

We introduce the variant `w/oALT`, where the Augmented Lorentz Transformation (ALT) is replaced by a simple multi-layer perceptron (MLP) for geometric standardization. Note that, ALT rigorously preserves the geometric consistency during the mapping between manifolds of different curvatures and dimensions. However, MLP lacks explicit geometric constraints and does not guarantee the preservation of manifold structures. As shown in Table 7, this variant (denoted as `w/oALT`) exhibits a consistent drop in performance across all benchmarks. The result suggests that MLP fails to capture the curved geometry, while ALT enables manifold-preserving representation under geometric transformations.

4.3.3 Cross-geometry attention vs. single self-attention

To assess the effectiveness of the proposed attention mechanism, we propose a variant of `w/oCGA`, which replaces the cross-geometry attention with conventional single-geometry attention. In particular, single-geometry attention considers the key, query and value in the same geometric factor. That is, different factors in the standardized product bundle do not interact with each other as in cross-geometry attention. As shown in Table 7, the `w/oCGA` variant shows a performance decrease compared to CRGFM, highlighting the crucial role of the attention mechanism in the transmission of geometric messages. In our design, the two factors in the product bundle are fused as a whole, improving the model expressiveness.

4.3.4 Curvature-based contrastive learning vs. graph autoencoder reconstruction

We design `w/oCCL` to isolate the impact of curvature-guided contrastive learning. To achieve this, we replace the self-supervised contrastive objective with a commonly used graph autoencoder loss, where a decoder reconstructs the adjacency matrix from latent embeddings. While both methods are unsupervised, the contrastive

Table 7 Ablation study on CRGFM variants for link prediction (AUC / AP in %). The best result in each column is bold

Model variant	Citeseer		Pubmed		Amazon-photo		Airport	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
`w/oMoGE`	99.10 ± 0.35	98.70 ± 0.40	95.20 ± 0.85	91.30 ± 0.55	96.00 ± 0.80	94.00 ± 0.45	94.30 ± 0.45	96.40 ± 0.50
`w/oALT`	99.15 ± 0.33	98.75 ± 0.38	95.25 ± 0.80	91.35 ± 0.58	96.10 ± 0.85	94.10 ± 0.50	94.40 ± 0.42	96.50 ± 0.46
`w/oCGA`	99.00 ± 0.37	98.60 ± 0.42	95.00 ± 0.90	91.20 ± 0.60	95.80 ± 0.88	93.90 ± 0.46	94.10 ± 0.48	96.20 ± 0.52
`w/oCCL`	99.18 ± 0.30	98.80 ± 0.36	95.30 ± 0.82	91.40 ± 0.57	96.20 ± 0.90	94.20 ± 0.44	94.50 ± 0.40	96.60 ± 0.43
CRGFM	**99.41 ± 0.29**	**99.01 ± 0.28**	**95.47 ± 0.95**	**91.58 ± 0.63**	**96.38 ± 0.91**	**94.29 ± 0.48**	**94.78 ± 0.39**	**96.75 ± 0.24**

loss exploits multiple geometric views from the product bundle, whereas the autoencoder focuses solely on topological proximity. As shown in Table 7, our contrastive loss enables geometric consistency by aligning representations from different curvatures through a unified tangent projection. In contrast, the autoencoding objective yields less transferable representations due to its emphasis on local reconstruction. These results highlight the advantage of using curvature-induced views and parallel transport alignment to improve the model expressiveness.

4.3.5 On geometric standardization

This part evaluates the proposed geometric standardization, and further discusses the difference to RiemannGFM. Here, we introduce a variant model named w/oGS removing the geometric standardization. To be specific, instead of creating a standardized product bundle as a whole, we leverage two bundles — a hyperbolic bundle of standard curvature -1 and a spherical bundle of standard curvature $+1$ — to serve as the representation space, and each bundle is associated with a pre-trained Riemannian transformer. Accordingly, the geometric experts of the non-spherical geometry are connected to the hyperbolic bundle, while non-hyperbolic geometric experts are connected to the spherical bundle. The curvature transformation between geometric expert and standardized bundle is done via exponential and logarithmic maps with a tangent space. We show the results in Fig. 3. Comparing with the proposed CRGFM, the variant of w/oGS presents a consistent performance drop on all datasets, suggesting the benefit of geometric standardization. In addition, CRGFM consistently outperforms RiemannGFM. Contrast to a handcrafted structural vocabulary in RiemannGFM, CRGFM is equipped with more flexible structural learning module of MoGE (mixture of geometric experts).

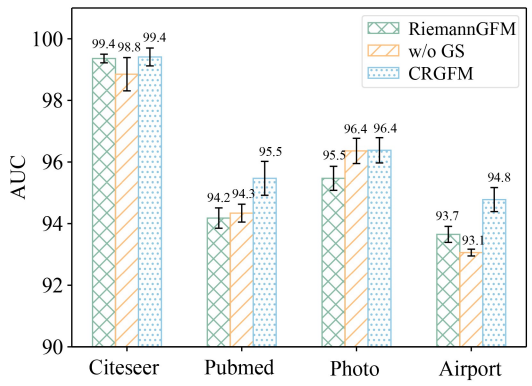


Fig. 3 The ablation results of geometric standardization

4.4 Domain difference and pre-training scales

This part studies how domain difference and pre-training scales affect the performance of the proposed CRGFM. To evaluate domain difference, we leverage different types of graph datasets for pre-training and testing. For example, CRGFM is pre-trained on Citeseer dataset, a citation network, and then is fine-tuned and tested on Airport dataset of different type, a collection of airline data. We summarize the result of node classification in Table 8, and we find that: 1) CRGFM shows robust cross-domain transferability that CRGFM is capable to generate promising node classification results even though it is pre-trained a graph of different type. 2) CRGFM exhibits superior performance when pre-training and testing datasets enjoy higher domain similarity. For example, when pre-trained on a citation network of Citeseer, CRGFM achieve 95.01 ± 0.75 in terms of AUC on Pubmed of the same type, but its AUC is shown to be 92.45 ± 0.58 on Airport dataset of different type. In other words, it shows a preference in cross-domain transferability that the performance will be increased when pre-training is conducted on the same type of graphs.

To examine the impact of pre-training scales, we increase the number of pre-training datasets from 1 to 4, and report the testing results in Table 9. Each target graph receives performance gain as the number of pre-training datasets as evidenced in Table 9. The results demonstrate that incorporating more pre-training datasets generally improves model performance.

4.5 Clustering and visualization

In this part, we evaluate the model performance on node clustering. To better understand the clustering results, we visualize the cluster confusion matrix in Fig. 4, where each row corresponds to a dataset and each column shows the output from a different model: GCN, DGI, GCOPE, Opengraph, and the proposed CRGFM. Also, we show the node embeddings using t-SNE in Fig. 5, where each point is colored by its ground-truth class label. The visualization on Cora,

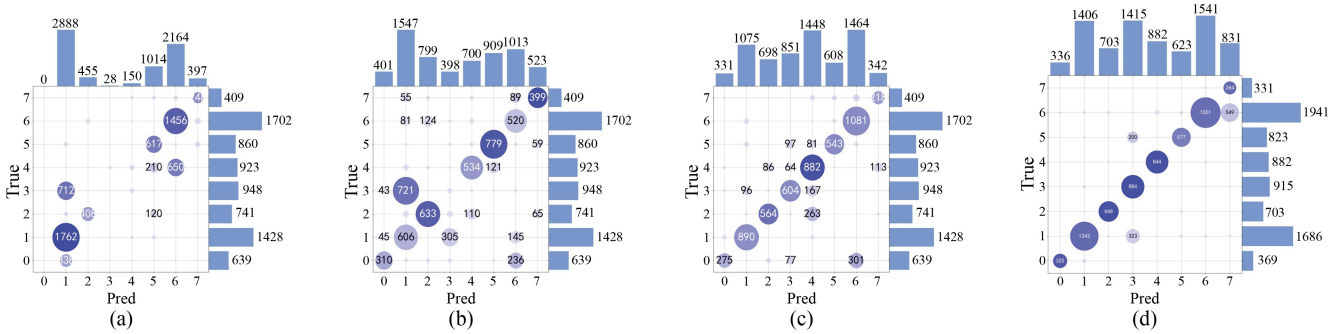


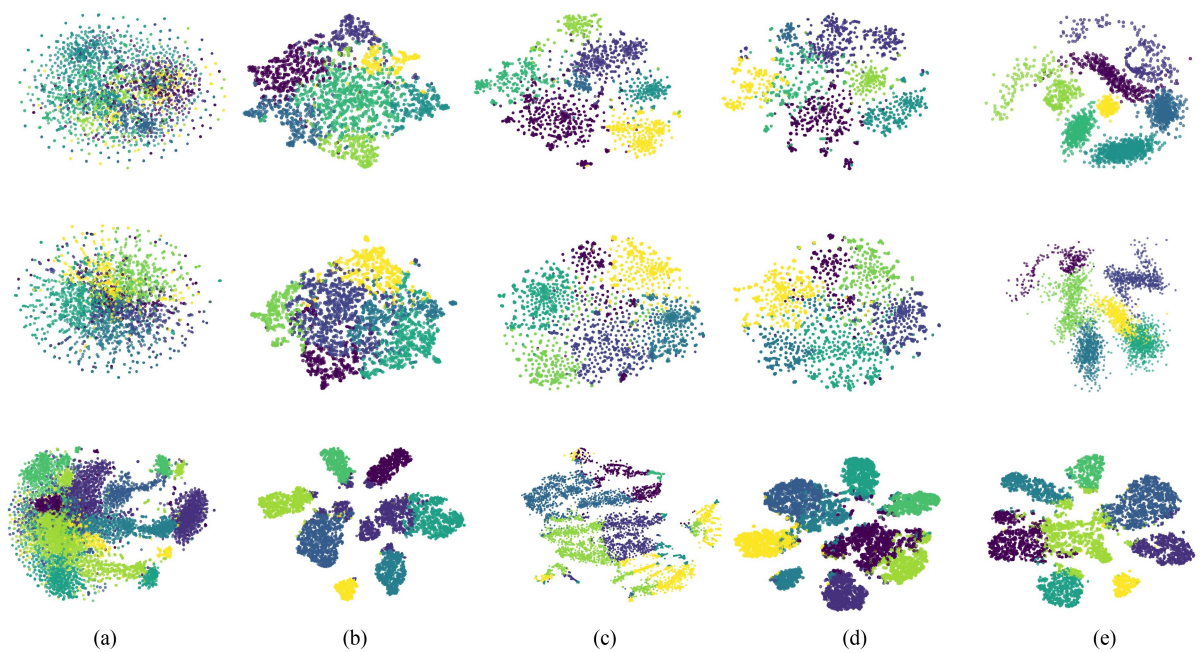
Fig. 4 Cluster confusion matrix of GCN, DGI, GCOPE, and CRGFM on the photo dataset. The vertical axis indicates how many nodes are contained in the true clusters, and the horizontal axis shows the number of nodes predicted by the model. (a) GCN; (b) DGI; (c) GCOPE; (d) CRGFM

Table 8 Performance in the transfer setting with different type of pre-training datasets in terms of AUC (%)

Pre-training	Testing datasets			
	Citeseer	Pubmed	Photo	Airport
Citeseer	**98.91 ± 0.19**	95.01 ± 0.75	95.45 ± 0.33	92.45 ± 0.58
Pubmed	98.34 ± 0.31	**95.33 ± 0.12**	95.69 ± 0.83	93.12 ± 0.89
Photo	96.37 ± 0.74	94.12 ± 0.50	**96.08 ± 0.21**	93.59 ± 0.58
Airport	96.11 ± 0.84	93.88 ± 0.67	95.12 ± 0.46	**94.14 ± 0.19**

Table 9 Performance in the transfer setting with different number of pre-training datasets in terms of AUC (%)

Pre-training	Testing datasets			
	Citeseer	Pubmed	Photo	Airport
Citeseer	98.91±0.19	95.01±0.75	95.45±0.33	92.45±0.58
Citeseer, Pubmed	98.56±0.62	94.54±0.19	95.77±0.81	92.85±0.15
Citeseer, Pubmed, Photo	99.31±0.29	95.44±0.95	96.25± 0.91	93.78±0.39
All datasets	99.18±0.54	95.43±0.77	96.19±0.76	94.55±0.72



(a) (b) (c) (d) (e)

Fig. 5 Visualization on Cora, CiteSeer, and Amazon-photo datasets. (a) Original; (b) GCN; (c) DGI; (d) GCOPE; (e) CRGFM

Citeseer and Amazon-photo demonstrates that embeddings generated by CRGFM exhibit superior class separability compared to all baselines. Notably, while vanilla GNNs (e.g., GCN) and self-supervised methods (e.g., DGI) tend to generate entangled or partially overlapping clusters, CRGFM yields more dispersed and well-isolated clusters, especially in the presence of structural irregularities. This distinction becomes even more evident when comparing CRGFM to recent graph foundation models like OpenGraph, suggesting that our approach effectively resolves geometric entanglement and enhances representation ability. That is, the Riemannian construction in CRGFM improves the structural understanding in the graph domain.

5 Related work

This section briefly reviews related work on graph representation learning, graph foundation models, and Riemannian deep learning.

5.1 Graph representation learning

In this decade, Graph Neural Networks (GNNs) have become the dominant solution for learning on graphs. Message-Passing Neural Networks (MPNNs) mark the initial success in graph deep learning.

Grounded on the spectral graph theory, early work defines the convolutions via graph Laplacian eigendecomposition [37]. The popular graph convolutional network [6] leverages the Chebyshev polynomial filters of the first order to achieve efficient spectral approximations. Meanwhile, [6] conducts neighborhood aggregation in the spatial domain. A series of subsequent efforts are devoted to improving the scalability and expressiveness. For example, GraphSAGE [5] is an inductive learning model with neighbor sampling, while GAT [30] leverages the attention mechanism to learn the pairwise weight among nodes. Recent advances on MPNN also reconsider the fundamental issues, such as over-smoothing [38] and over-squashing [39,40]. Another line of work is the Graph Transformer, which extends the global attention of classic transformer on graphs. Pioneering works such as [21] integrated positional encodings to preserve structural information in the transformer architectures. Further advances incorporated explicit structural biases, e.g., Graphormer [41] used centrality and spatial encoding to improve attention scores. These models typically excel in tasks requiring long-range dependency capture. Either MPNNs or graph transformers conduct end-to-end training with the target learning task. Such task-specific graph models face the shortcoming of task transferability. Hence, recent years have witnessed the marriage of graph deep learning and self-supervised learning [42–44]. For example, graph contrastive learning, which explores the similarity of data themselves, decouples model training from target learning task [9], but graph augmentation for contrastive learning is not trivial [45]. Also, the gap between self-supervised learning pretext and target task tends to limit the model expressiveness. Given the structural diversity in the graph domain, this line of work cannot handle the graph transferability. Limitations in graph and task transferability motivate the graph foundation model.

5.2 Graph foundation models

Graph foundation models evolve along two primary pathways: GNN-based methods and LLM-based methods. The former enhances the transferability by adaptation mechanisms such as unifying downstream tasks via graph prompting and fine-tuning external task-specific layers. GFT [46] abstracts transferable patterns into computation tree vocabularies to decouple local motif structures. ProNoG [47] employs conditional networks to generate node-specific prompts for non-homophilic graphs, while EdgePrompt [48] integrates learnable edge prompt vectors into message passing to enhance structural information transfer. RiemannGFM [22] introduces the notion of structural vocabulary to unify graph modeling over a shared collection of substructures. GNN-based methods exhibit fragility under structural shifts across domains due to structural biases in model pre-training. As for LLM-based methods, LLaGA [49] transforms nodes into structure-aware sequences for LLM-based end-to-end task prediction. UniGraph [50] achieves cross-domain transferability via cascaded LM-GNN architectures, while GraphAlign [51] aligns feature distributions through Mixture-of-Experts with dynamic projectors for unified multi-graph pretraining. LLM-based methods primarily rely on the sequential description of graphs, which deconstructs the structural complexity of graphs. Distinguishing from existing GFMs, we emphasize both structural complexity and structural diversity in the graph domain, and introduce the curvature-based description and standardization of Riemannian geometry for the first time, to the best of our knowledge.

5.3 Riemannian graph learning

Riemannian geometry offers a systematic construction for structural analysis, and has shown superiority in graph representation learning in recent years. Hyperbolic space is first recognized for its expressiveness of tree-like/hierarchical structures, and a diversity of hyperbolic GNNs are formulated, e.g., HGNN [3] and HGCN [29]. Hyperspherical spaces are adept at representing cyclic or symmetric structures, such as rings and cycles [19,52]. So far, Riemannian graph learning often considers geodesically complete manifolds for effective optimization. Other examples include product manifolds [25], the κ-stereographic model [27] and pseudo-Riemannian manifold under certain formulation [28]. Recently, the concept of Mixture-of-Experts has been introduced for personalized geometric matching. Beyond graph embedding, [27,53] leverages the Riemannian manifold and its geometric tools for node clustering. Another line of work considers graph generation with Riemannian geometry, e.g., [4] extends the denoising diffusion model to the product space, and [54] builds the structural Schrödinger bridge on the manifold. The majority of Riemannian graph learning primarily focuses on studying the geometry of specific graphs, and thus is orthogonal to the universality of foundation models. Very recently, RiemannGFM [22] introduced the Riemannian geometry to GFM. However, it explicitly samples the tree and cycle substructures to align with its factor component, while we implicitly model the tree-like and cyclical patterns in the standardized product bundle. Additionally, we propose the first Riemannian prompt learning for GFM to the best of our knowledge.

6 Conclusion

In this paper, we reconsidered the structural complexity and diversity in the graph domain, and presented a geometric graph foundation model CRGFM, which explores the universality and expressiveness via geometric transformations in Riemannian geometry. In particular, we addressed the structural complexity through a novel standardization phase of augmented Lorentz transformations, while modeling the structural complexity in the latent product bundle by the cross-geometry Riemannian graph transformer. Furthermore, we conducted the parameterized manifold perturbation (prompt learning) to bridge the gap between model pre-training and target learning tasks. Extensive experiments on real-world graphs demonstrate the superior cross-domain transferability of CRGFM in few-shot learning and zero-shot learning, and its effectiveness in a wide range of downstream tasks.

Broader impact. This work connects two research realms of graph and geometry, and shows the potential of a universal deep network for the graph domain. Encouraged by the tremendous success of LLMs, there has recently been a surge of interest in studying the graph foundation models, where we highlight the significance of geometric analysis over graphs. A positive societal impact lies in the

energy consuming re-training of graph neural network can be reduced, while maintaining the expressiveness to new graphs in reality. None of negative societal impacts we feel must be specifically highlighted.

Future direction. Possible future directions include expanding the universality along with the mission of graph foundation model. Also, we notice that the existing GFMs including ours primarily focus on the undirected graphs, while there exists real-world graphs that are directly by nature.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62202164). Philip S. Yu is supported in part by NSF under grants III-2106758, and POSE-2346158.

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendixes

Implementation details

In the proposed CRGFM, we do not optimize the manifold-valued node representations, and the model parameters reside in the tangent spaces. Thus, we employ the popular Adam optimizer for parameter training, where the learning rate is $3e-5$ and the dropout rate is 0.3. We undergo the model training of 5000 epochs by default. Our model is build upon PyTorch (at the website of pytorch.org) and Geoopt (at the website of github.com/geoopt/). As for the hyperparameters, we set the weighing coefficient between the embedding distortion loss and contrastive learning loss as 1. That is, the loss of embedding distortion and the loss of contrastive learning contribute equally in the proposed CRGFM. The number of geometric experts is 11. More specifically, there are 5 hyperbolic geometric experts, 1 Euclidean geometric experts and 5 spherical geometric experts. The curvature of Euclidean geometric expert is zero, while those of hyperbolic/spherical geometric experts are learnable, and are fine-tuned according to the target learning task.

References

[1] Wei C, Liang J, Liu D, Wang F. Contrastive graph structure learning via information bottleneck for recommendation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1484

[2] Li A, Yang B. Dual graph denoising model for social recommendation. In: Proceedings of the ACM on Web Conference 2025. 2025, 347–356

[3] Liu Q, Nickel M, Kiela D. Hyperbolic graph neural networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019, 739

[4] Wang Y, Zhang S, Ye J, Peng H, Sun L. A mixed-curvature graph diffusion model. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024, 2482–2492

[5] Hamilton W L, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 1025–1035

[6] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations. 2017, 1–14

[7] Kreuzer D, Beaini D, Hamilton W L, Létourneau V, Tossou P. Rethinking graph transformers with spectral attention. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 1654

[8] Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J. Open graph benchmark: datasets for machine learning on graphs. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1855

[9] Zhu Y, Xu Y, Yu F, Liu Q, Wu S, Wang L. Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021. 2021, 2069–2080

[10] Hou Z, He Y, Cen Y, Liu X, Dong Y, Kharlamov E, Tang J. GraphMAE2: a decoding-enhanced masked self-supervised graph learner. In: Proceedings of the ACM Web Conference. 2023, 737–746

[11] Jin W, Liu X, Zhao X, Ma Y, Shah N, Tang J. Automated self-supervised learning for graphs. In: Proceedings of the 10th International Conference on Learning Representations. 2022, 1–20

[12] Xia L, Huang C. AnyGraph: graph foundation model in the wild. 2024, arXiv preprint arXiv: 2408.10700

[13] OpenAI. GPT-4 technical report. 2023, arXiv preprint arXiv: 2303.08774

[14] Mao H, Chen Z, Tang W, Zhao J, Ma Y, Zhao T, Shah N, Galkin M, Tang J. Position: graph foundation models are already here. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1410

[15] Tang J, Yang Y, Wei W, Shi L, Su L, Cheng S, Yin D, Huang C. GraphGPT: graph instruction tuning for large language models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 491–500

[16] Liu Z, Yu X, Fang Y, Zhang X. GraphPrompt: unifying pre-training and downstream tasks for graph neural networks. In: Proceedings of the ACM Web Conference 2023. 2023, 417–428

[17] Zhang M, Sun M, Wang P, Fan S, Mo Y, Xu X, Liu H, Yang C, Shi C. GraphTranslator: aligning graph model to large language model for open-ended tasks. In: Proceedings of the ACM Web Conference 2024. 2024, 1003–1014

[18] Shchur O, Mumme M, Bojchevski A, Günnemann S. Pitfalls of graph neural network evaluation. 2018, arXiv preprint arXiv: 1811.05868

[19] Petersen P. Riemannian Geometry. 3rd ed. Cham: Springer, 2016

[20] Gu A, Sala F, Gunel B, Ré C. Learning mixed-curvature representations in product spaces. In: Proceedings of the 7th International Conference on Learning Representations. 2019, 1–21

[21] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000–6010

[22] Sun L, Huang Z, Zhou S, Wan Q, Peng H, Yu P. RiemannGFM: learning a graph foundation model from Riemannian geometry. In: Proceedings of the ACM on Web Conference 2025. 2025, 1154–1165

[23] Chami I, Wolf A, Juan D C, Sala F, Ravi S, Ré C. Low-dimensional hyperbolic knowledge graph embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020, 6901–6914

[24] Sun L, Zhang Z, Zhang J, Wang F, Peng H, Su S, Yu P S. Hyperbolic variational graph neural network for modeling dynamic graphs. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. 2021, 4375–4383

[25] de Ocáriz Borde H S, Kazi A, Barbero F, Liò P. Latent graph inference using product manifolds. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–36

[26] Sun L, Huang Z, Wang Z, Wang F, Peng H, Yu P S. Motif-aware Riemannian graph neural network with generative-contrastive learning. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 9044–9052

[27] Bachmann G, Bécigneul G, Ganea O E. Constant curvature graph convolutional networks. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 46

[28] Xiong B, Zhu S, Potyka N, Pan S, Zhou C, Staab S. Pseudo-Riemannian graph convolutional networks. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 252

[29] Chami I, Ying Z, Re C, Leskovec J. Hyperbolic graph convolutional neural networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019, 438

[30] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations. 2018, 1–12

[31] Yang M, Verma H, Zhang D C, Liu J, King I, Ying R. Hypformer: exploring efficient transformer fully in hyperbolic space. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 3770–3781

[32] Yang Z, Cohen W W, Salakhutdinov R. Revisiting semi-supervised learning with graph embeddings. In: Proceedings of the 33rd International Conference on Machine Learning. 2016, 40–48

[33] Veličković P, Fedus W, Hamilton W L, Liò P, Bengio Y, Hjelm R D. Deep graph infomax. In: Proceedings of the 7th International Conference on Learning Representations. 2019, 1–17

[34] Zhao H, Chen A, Sun X, Cheng H, Li J. All in one and one for all: a simple yet effective method towards cross-domain graph pretraining. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 4443–4454

[35] Liu H, Feng J, Kong L, Liang N, Tao D, Chen Y, Zhang M. One for all: towards training one graph model for all classification tasks. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–23

[36] Xia L, Kao B, Huang C. OpenGraph: towards open graph foundation models. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024. 2024, 2365–2379

[37] Yu T, Sa C D. Random Laplacian features for learning with hyperbolic space. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–23

[38] Oono K, Suzuki T. On asymptotic behaviors of graph CNNs from dynamical systems perspective. 2019, arXiv preprint arXiv: 1905.10947

[39] Alon U, Yahav E. On the bottleneck of graph neural networks and its practical implications. In: Proceedings of the 9th International Conference on Learning Representations. 2021, 1–16

[40] Sun L, Huang Z, Wu H, Ye J, Peng H, Yu Z, Yu P S. DeepRicci: self-supervised graph structure-feature co-refinement for alleviating over-squashing. In: Proceedings of 2023 IEEE International Conference on Data Mining. 2023, 558–567

[41] Yang J, Liu Z, Xiao S, Li C, Sun G, Xie X. GraphFormers: GNN-nested language models for linked text representation. 2021, arXiv preprint arXiv: 2105.02605

[42] Sun L, Ye J, Peng H, Yu P S. A self-supervised Riemannian GNN with time varying curvature for temporal graph learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022, 1827–1836

[43] Sun L, Zhang Z, Ye J, Peng H, Zhang J, Su S, Yu P S. A self-supervised mixed-curvature graph neural network. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence. 2022, 4146–4155

[44] Sun L, Ye J, Peng H, Wang F, Yu P S. Self-supervised continual graph learning in adaptive Riemannian spaces. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 4633–4642

[45] Ju W, Wang Y, Qin Y, Mao Z, Xiao Z, Luo J, Yang J, Gu Y, Wang D, Long Q, Yi S, Luo X, Zhang M. Towards graph contrastive learning: a survey and beyond. 2024, arXiv preprint arXiv: 2405.11868

[46] Wang Z, Zhang Z, Chawla N V, Zhang C, Ye Y. GFT: graph foundation model with transferable tree vocabulary. In: Advances in Neural Information Processing Systems 38. 2024

[47] Yu X, Zhang J, Fang Y, Jiang R. Non-homophilic graph pre-training and prompt learning. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1. 2025, 1844–1854

[48] Ning J, Zhu P, Zheng C, Lee G, Sun S, Yang T. EdgePrompt: a distributed key-value inference framework for LLMs in 6g networks. In: Proceedings of the IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2025, 1–6

[49] Chen R, Zhao T, Jaiswal A, Shah N, Wang Z. LLaGA: large language and graph assistant. In: Proceedings of the 41st International

Conference on Machine Learning. 2024, 306
[50] He Y, Sui Y, He X, Hooi B. UniGraph: learning a unified cross-domain foundation model for text-attributed graphs. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1.2025, 448-459
[51] Song Z, Wei H, Bai L, Yang L, Jia C. GraphAlign: enhancing accurate feature alignment by graph matching for multi-modal 3D object detection. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 3335–3346
[52] Tan C, Gao Z, Wu L, Li S, Li S Z. Hyperspherical consistency regularization. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 7234–7245
[53] Sun L, Hu J, Zhou S, Huang Z, Ye J, Peng H, Yu Z, Yu P. RicciNet: deep clustering via a Riemannian generative model. In: Proceedings of the ACM Web Conference 2024. 2024, 4071–4082
[54] Sun L, Zhou S, Fang B, Zhang H, Ye J, Ye Y, Yu P S.Trace: Structural Riemannian Bridge Matching for Transferable Source Localization in Information Propagation In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence. 2025, 3308–3316

Li SUN is currently an associate professor at Beijing University of Posts and Telecommunications, China. He received his B.S. and Ph.D. degrees in computer science from Beijing University of Posts and Telecommunications, China. He has been a visiting scholar at University of Illinois Chicago, USA, advised by Prof. Philip S. Yu. His research interests lie in deep learning and data mining with special attentions to graphs and Riemannian geometry. He is the recipient of ACM CIKM 2022 best paper honorable mention. He has published over 50 papers in referred conferences and journals including ICML, NeurIPS, KDD, WWW, AAAI, TOIS, TKDE, TIST, and TWEB.

Philip S. YU (Fellow, ACM/IEEE) received his BS degree in EE from Taiwan University, China, his MS and PhD degrees in EE from Stanford University, USA. He is a distinguished professor in computer science with the University of Illinois Chicago, USA and also holds the Wexler chair in information technology. He is the recipient of the ACM SIGKDD 2016 Innovation Award, the IEEE Computer Society 2013 Technical Achievement Award, the IEEE ICDM 2003 Research Contributions Award, IEEE 1999 Region 1 Award, PAKDD 2025 Distinguished Research Contributions Award, VLDB 2022 Test of Time Award, ACM SIGSPATIAL 2021 10-year Impact Award, the EDBT 2014 Test of Time Award, the ICDM 2013 10-year Highest Impact Paper Award, and SDM 2008 Best Paper Award. He has served as program chair or co-chairs for DASFAA'25, ICDE'95, etc., and general chair or co-chairs for CIKM'06, ICDM'02, ICDE'99, etc. Philip S. Yu has published more than 1,700 referred conference and journal papers cited more than 226,000 times with an H-index of 207. He was the Editor-in-Chiefs for ACM Transactions on Knowledge Discovery from Data (2011−2017) and IEEE Transactions on Knowledge and Data Engineering (2001−2004).