Adaptive Riemannian Graph Neural Networks

Xudong Wang, Chris Ding, Tongxin Li, Jicong Fan*

School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
xudongwang@link.cuhk.edu.cn, {chrisding, litongxin, fanjicong}@cuhk.edu.cn

Abstract

Graph data often exhibits complex geometric heterogeneity, where structures with varying local curvature, such as tree-like hierarchies and dense communities, coexist within a single network. Existing geometric GNNs, which embed graphs into single fixed-curvature manifolds or discrete product spaces, struggle to capture this diversity. We introduce Adaptive Riemannian Graph Neural Networks (ARGNN), a novel framework that learns a *continuous and anisotropic Riemannian metric tensor field over the graph*. It allows each node to determine its optimal local geometry, enabling the model to fluidly adapt to the graph's structural landscape. Our core innovation is an efficient parameterization of the node-wise metric tensor, specializing to a learnable diagonal form that captures directional geometric information while maintaining computational tractability. To ensure geometric regularity and stable training, we integrate a Ricci flow-inspired regularization that smooths the learned manifold. Theoretically, we establish the rigorous geometric evolution convergence guarantee for ARGNN and provide a continuous generalization that unifies prior fixed or mixed-curvature GNNs. Empirically, our method demonstrates superior performance on both homophilic and heterophilic benchmark datasets with the ability to capture diverse structures adaptively. Moreover, the learned geometries both offer interpretable insights into the underlying graph structure and empirically corroborate our theoretical analysis.

1 Introduction

Real-world graphs, from social networks to protein interaction maps, exhibit a rich geometric diversity that challenges various graph learning paradigms (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017; Sun, Ding, and Fan 2023; Kang et al. 2023; Wang and Fan 2024; Sun et al. 2024; Sun and Fan 2024; Wang et al. 2025b; Qian et al. 2025; Wang et al. 2025c; Guo et al. 2025; Fan 2025a,b). Consider a social network where some communities form deep, tree-like hierarchies best captured by hyperbolic geometry (Chami et al. 2019), while others create dense, tightly-knit cliques that resemble spherical manifolds (Gu et al. 2018). Forcing such a geometrically heterogeneous graph into a single geometric space (Euclidean, hyperbolic, or spherical) inevitably introduces significant distortion and information loss.

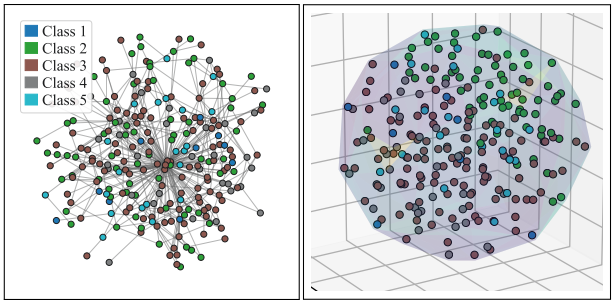

Figure 1: **Geometric heterogeneity in WISCONSIN network. Left:** raw graph topology coloured by class. **Right:** 3-D t-SNE of node features. The translucent hull is coloured by the magnitude of discrete mean curvature (violet → flat, yellow → strongly curved), showing that curvature varies across the *Riemannian manifold*.

Pioneering work in geometric Graph Neural Networks (GNNs) has demonstrated the power of non-Euclidean spaces. Hyperbolic GNNs (Chami et al. 2019; Zhang et al. 2021) excel on tree-like graphs but struggle with dense cycles. Conversely, spherical methods handle cyclic structures well but are less effective on hierarchical data. This fundamental limitation stems from their shared assumption of *global geometric homogeneity*.

Recent advances have moved towards mixed-curvature approaches to address this issue. Models like κ-GCN (Bachmann, Bécigneul, and Ganea 2020) learn an isotropic (scalar) curvature for each node, while state-of-the-art methods like CUSP (Grover et al. 2025) embed graphs into product manifolds of constant-curvature spaces (e.g., $\mathbb{H}^k \times \mathbb{S}^l \times \mathbb{E}^m$). While a significant step forward, these methods remain constrained. Scalar curvature approaches are still isotropic at the node level, unable to capture directional geometric information. Product manifold methods are limited to block-constant geometries chosen from a discrete set of curvatures. Neither can fully capture the fine-grained, continuous geometric variations inherent in complex data.

This geometric heterogeneity is not merely a theoretical construct; it is evident in real-world data, as illustrated in

*Corresponding author.

Figure 1. The real-world WISCONSIN network (Craven et al. 2000) embeds into a Riemannian feature manifold whose neighbourhoods can differ sharply in both curvature and orientation. Any single or fixed-curvature space would therefore distort part of the data, motivating our node-wise, adaptive metric tensor.

The importance of geometric understanding has been further highlighted by recent advances in curvature-based graph analysis. Graph Neural Ricci Flow (GNRF) (Chen et al. 2025) demonstrates that evolving features according to discrete Ricci curvature improves representation quality, though it keeps the underlying geometry fixed. Discrete curvature methods like CurvDrop (Liu et al. 2023) and curvature-based graph rewiring (Topping et al. 2021) show promise but operate as preprocessing steps rather than end-to-end learning frameworks.

Our key insight is to move beyond pre-defined geometries by learning a **continuous, node-adaptive Riemannian metric field** that directly models the local structure. To realize this vision within a tractable framework, we introduce **Adaptive Riemannian Graph Neural Networks** (ARGNN). Our framework proposes a principled and efficient parameterization of the metric tensor. Specifically, we model each node's local geometry with an *anisotropic diagonal metric*, $\mathbf{G}_i = \mathrm{diag}(\mathbf{g}_i)$. This design is not merely a computational shortcut; it corresponds to a flexible, per-dimension scaling of the feature space geometry. To ensure stable learning, we further integrate Ricci flow-inspired dynamics that regularize the evolution of the learned geometric manifold. See the full version (Wang et al. 2025a) for appendices with proofs, ablations, and implementation details. The contributions of our work are threefold:

i) **Methodologically**, we propose a principled parameterization of the adaptive metric field using diagonal matrices. We provide strong geometric and algorithmic justifications for our design, framing it as an anisotropic conformal transformation that is both computationally efficient and highly expressive. To the best of our knowledge, ARGNN is the first framework that learns continuous and anisotropic metric tensor fields for graphs.

ii) **Theoretically**, we establish the convergence guarantee for ARGNN's geometric evolution and further prove its role as a universal framework that generalizes and unifies prior isotropic, scalar-curvature, and product-manifold GNNs.

iii) **Empirically**, we conduct extensive experiments on a wide range of benchmark datasets, including comprehensive ablation studies. Our results demonstrate that ARGNN achieves superior performance compared to state-of-the-art benchmark methods on both homophilic and heterophilic graphs.

2 Related Work

2.1 Fixed-Curvature Geometric GNNs

Geometric deep learning on graphs has been revolutionized by embedding approaches that leverage non-Euclidean geometries. **Hyperbolic GNNs** exploit the exponential volume growth of hyperbolic space to naturally represent hierarchical structures. HGCN (Chami et al. 2019) extends graph convolutions to hyperbolic space using the Poincaré ball model, achieving remarkable success on tree-like graphs. HGAT (Zhang et al. 2021) incorporates attention mechanisms in hyperbolic space, while recent work explores hyperbolic transformers (Gulcehre et al. 2018) and variational autoencoders (Sun et al. 2021).

Spherical GNNs address the complementary challenge of cyclic and community structures. Spherical CNNs (Cohen et al. 2018) and their graph extensions (Gu et al. 2018) leverage the positive curvature of spherical space to model dense, interconnected communities. However, both hyperbolic and spherical approaches assume global geometric homogeneity, limiting their applicability to mixed-topology graphs.

Fixed-curvature methods have the advantage compared with traditional GNNs in different target domains but fail catastrophically when graph geometry mismatches the embedding space, as shown in Figure 1, which prevents them from effectively handling the geometric heterogeneity prevalent in real-world networks.

2.2 Discrete Mixed-Curvature Approaches

Recognizing the limitations of fixed-curvature methods, recent work explores mixed-curvature embeddings.

κ-GCN (Bachmann, Bécigneul, and Ganea 2020) learns scalar curvature parameters for each node, allowing adaptation between hyperbolic, Euclidean, and spherical geometries. However, scalar curvature cannot capture directional geometric information.

CUSP (Grover et al. 2025) represents the current state-of-the-art in mixed-curvature approaches. It combines spectral graph analysis with embeddings in product manifolds of constant-curvature spaces (e.g., $\mathbb{H}^k \times \mathbb{S}^l \times \mathbb{E}^m$). CUSP also introduces spectral filtering techniques and curvature-aware graph Laplacians, achieving strong performance across diverse benchmarks.

$\mathcal{Q}$-GCN (Xiong et al. 2022) extends the framework to pseudo-Riemannian manifolds with indefinite metrics, enabling more flexible curvature combinations. Self-supervised mixed-curvature methods (Jin et al. 2020) explore representation learning without labels.

However, despite their flexibility, these approaches primarily treat geometry as *discrete choices* from finite sets of constant-curvature manifolds. Consequently, they offer limited support for continuous geometric variation and anisotropic (directional) structure.

2.3 Curvature-Based Graph Analysis

Recent advances in discrete differential geometry have enabled sophisticated graph analysis through curvature.

Discrete Ricci curvature, including Ollivier-Ricci (Ollivier 2009) and Forman-Ricci (Forman 2003) curvature, provide local geometric characterizations of graph structure.

Graph Neural Ricci Flow (GNRF) (Chen et al. 2025) represents a significant recent advance. GNRF evolves node features according to discrete Ricci curvature, showing that curvature-aware feature evolution improves representation

quality. However, GNRF keeps the underlying graph geometry fixed while evolving features.

Curvature-based rewiring methods (Topping et al. 2021; Fesser and Weber 2024) use curvature analysis to identify and modify graph bottlenecks. CurvDrop (Liu et al. 2023) employs Ricci curvature for topology-aware dropout sampling. These methods show curvature's utility but operate as preprocessing steps rather than end-to-end learning.

Curvature-based methods have found success in protein analysis (Wu et al. 2023; Shen et al. 2024), community detection (Ni et al. 2019), and graph generation (Li et al. 2022). However, to our knowledge, *joint learning* of geometry and features remains rare; methods typically fix geometry and evolve features (GNRF) or use curvature for preprocessing.

3 Preliminaries

3.1 Riemannian Geometry Essentials

A **Riemannian manifold** $(\mathcal{M}, \mathbf{g})$ consists of a smooth manifold $\mathcal{M}$ equipped with a metric tensor field $\mathbf{g}$ that varies smoothly across the manifold (Lee 2018). At each point $p \in \mathcal{M}$, the metric tensor $\mathbf{g}_p$ defines an inner product on the tangent space $\mathcal{T}_p\mathcal{M}$, enabling the measurement of distances, angles, and curvatures.

For graph-structured data, we introduce a **metric tensor field over the graph** by associating each node $i \in \mathcal{V}$ with a metric tensor $\mathbf{G}_i \in \mathcal{S}^d_{++}$, where $\mathcal{S}^d_{++}$ denotes the cone of symmetric positive definite matrices. This creates a continuous geometric structure over the discrete graph domain and makes Christoffel symbols vanish, where each node's feature neighborhood in $\mathbb{R}^d$ possesses its own Riemannian geometry. The **geodesic distance** between sufficiently close points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ under the metric $\mathbf{G}$ of $\mathcal{T}_\mathbf{x}\mathcal{M}$ is:

$$d_\mathbf{G}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{G} (\mathbf{x}-\mathbf{y})}$$

The **Ricci curvature tensor** $\mathrm{Ric}(\mathbf{G})$ characterizes how the metric changes across the manifold. In differential geometry, **Ricci flow** (Chow and Knopf 2004) evolves a metric tensor field $\{\mathbf{G}_i(t)\}$ according to $\frac{\partial \mathbf{G}_i}{\partial t} = -2\mathrm{Ric}(\mathbf{G}_i)$, which smooths geometric irregularities while preserving topological structure, providing a principled approach for geometric regularization on graphs (Hamilton 1982). Due to space limitations, more Riemannian Geometry and Ricci curvature details are shown in Appendix B.

3.2 GNNs and Geometric Message Passing

Consider an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with nodes $\mathcal{V}$, edges $\mathcal{E}$, and node features $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. Standard message passing (Gilmer et al. 2017) of GNNs updates node representations through:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\big(\mathbf{W}_s^{(\ell)} \mathbf{h}_i^{(\ell)} + \sum_{j \in \mathcal{N}(i)} \mathbf{W}_m^{(\ell)} \mathbf{h}_j^{(\ell)}\big)$$

where $\mathcal{N}(i)$ denotes the neighborhood of node i, $\mathbf{W}_s^{(\ell)}$ and $\mathbf{W}_m^{(\ell)}$ are learnable transformation matrices (parameters) on ℓ-th layer, and σ is a nonlinear activation function.

Geometric message passing enhances this by incorporating Riemannian structure by embedding graphs in non-Euclidean spaces. Existing approaches embed graphs in fixed geometries (hyperbolic, spherical) or discrete mixed-curvature spaces. However, as illustrated in Figure 1, real graphs exhibit **continuous geometric heterogeneity** and the local geometry varies across the network.

3.3 Problem Formulation and Challenges

The core challenge of this work is to move beyond the fixed-geometry paradigm. We aim to learn a node-adaptive geometric space jointly with the node representations $\mathbf{H} = \{\mathbf{h}_i\}_{i \in \mathcal{V}}$. This learned space is mathematically formulated as a Riemannian metric tensor field $\{\mathbf{G}_i \in \mathcal{S}^d_{++}\}_{i \in \mathcal{V}}$, where each node i is endowed with its own local metric $\mathbf{G}_i$. However, realizing this vision presents several key hurdles.

First, from a mathematical standpoint, each learned tensor $\mathbf{G}_i$ must remain symmetric positive definite (SPD) throughout training to constitute a valid Riemannian metric. **Second**, from a computational perspective, parameterizing a full $d \times d$ metric tensor for every node is prohibitive, scaling as $O(|\mathcal{V}|d^2)$ and hindering application to large graphs. **Third**, for learning stability, the geometric manifold must evolve smoothly; pathological curvatures could destabilize the training process and lead to poor generalization. **Finally**, to justify its increased complexity, the proposed framework must be provably more expressive than its fixed-geometry predecessors. Our ARGNN framework is designed to address these challenges systematically.

4 Adaptive Riemannian Graph Neural Networks

This section introduces Adaptive Riemannian Graph Neural Networks (ARGNN), a framework that learns a continuous, node-adaptive geometry tailored to the underlying graph structure. Our approach forgoes the manual selection of a geometric space and instead learns the geometry as part of the end-to-end training process. This section details the three core components of our framework: i) an efficient and interpretable parameterization of the node-wise metric tensor via a local metric estimator, ii) a geometric message passing scheme that leverages the learned metric, and iii) a Ricci flow-inspired regularization for stable geometric evolution. Figure 2 illustrates our approach's diagram.

4.1 Learning an Anisotropic Metric Field

The foundational concept of our work is to equip the graph with a **Riemannian metric tensor field**, denoted as $\{\mathbf{G}_i \in \mathcal{S}^d_{++}\}_{i \in \mathcal{V}}$. Each metric tensor $\mathbf{G}_i$ is a symmetric positive-definite (SPD) matrix that defines a unique local geometry on the feature space neighborhood of node i. As stated before, a direct parameterization of a full $d \times d$ matrix for each node will be computationally prohibitive and lead to expensive downstream computations.

To address this, we propose a **principled simplification** by parameterizing each metric tensor $\mathbf{G}_i$ as a learnable **diagonal matrix**:

$$\mathbf{G}_i = \mathrm{diag}(\mathbf{g}_i) = \mathrm{diag}(g_{i,1}, g_{i,2}, \ldots, g_{i,d}) \tag{1}$$

where $\mathbf{g}_i \in \mathbb{R}^d_{++}$ is a vector of positive diagonal elements. This diagonal parameterization is not an arbitrary choice but a motivated design with several key advantages:

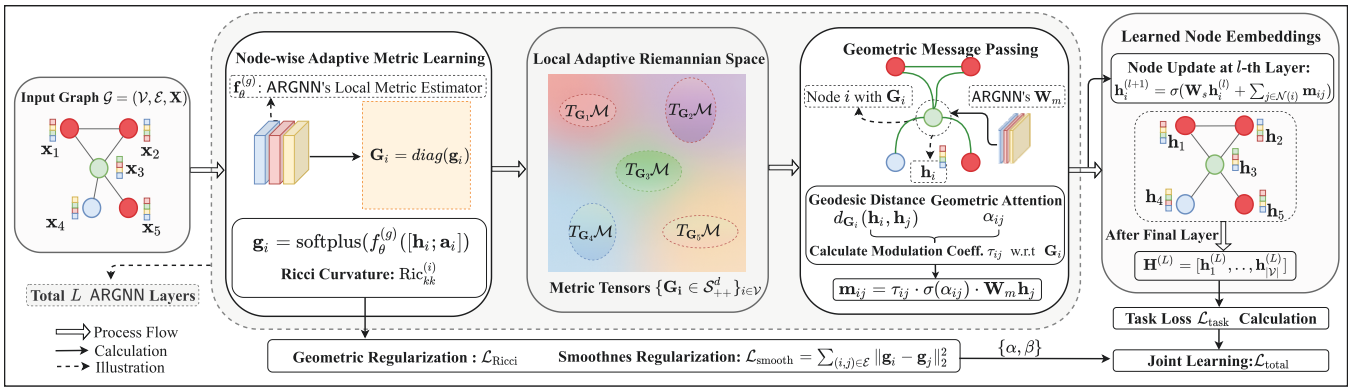

Figure 2: Diagram of our proposed ARGNN, which jointly learns continuous, anisotropic metric tensor fields $\{\mathbf{G}_i \in \mathcal{S}_{++}^d\}_{i\in\mathcal{V}}$ and node embeddings $\mathbf{H} = \{\mathbf{h}_i\}_{i\in\mathcal{V}}$. The learned $\mathbf{G}_i$ gives beyond curvature information to depict the geometric diversity.

i) **Geometric Interpretation:** At its core, a metric tensor defines how to measure distances, angles, and curvatures. The standard Euclidean space, governed by the identity matrix, provides a single, global way of measurement. Our diagonal metric, $\mathbf{G}_i = \text{diag}(\mathbf{g}_i)$, acts as a local, learned modification to this standard. Specifically, it *rescales the geometry independently along each feature axis*. The value of each diagonal element $g_{i,k}$ determines the stretch of the k-th dimension in the local vicinity of node i. This axis-aligned, non-uniform scaling type is formally known as an **anisotropic conformal transformation** (Obata 1970; Spivak 1999). It endows the model with far greater flexibility than the isotropic scaling of constant-curvature spaces, while remaining more structured and tractable than a full metric tensor. The formal details of this geometric perspective are elaborated in Appendix B.

ii) **Interpretability and Feature Decomposition:** The diagonal form assumes that the standard coordinate axes of the feature space align with the principal directions of the local geometry. This makes the model highly interpretable: each diagonal element $g_{i,k}$ directly quantifies the geometric importance or local scaling factor of the k-th feature dimension for node i. This is particularly effective when features are learned to be relatively disentangled.

iii) **Computational and Algorithmic Efficiency:** This parameterization dramatically reduces the complexity of learning and using the metric. The number of geometric parameters per node drops from $O(d^2)$ to $O(d)$. Furthermore, crucial geometric computations such as geodesic distance, matrix inversion, and eigendecomposition become highly efficient, which we will discuss later.

To learn these adaptive metrics, we employ a small neural network $f_\theta^{(g)}$ as the local metric estimator that maps a node's local structural information to its metric vector $\mathbf{g}_i$. Specifically, for each node i, we first aggregate its neighborhood features $\mathbf{a}_i = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \mathbf{h}_j$, and then compute its metric vector as:

$$\mathbf{g}_i = \text{softplus}\left(f_\theta^{(g)}([\mathbf{h}_i; \mathbf{a}_i])\right) \tag{2}$$

where $[\cdot;\cdot]$ denotes feature concatenation. The softplus activation function ensures all elements of $\mathbf{g}_i$ are strictly positive, thereby naturally satisfying the SPD constraint for $\mathbf{G}_i$.

4.2 Geometric Message Passing

With the learned diagonal metric field, we now formulate a message passing scheme that is explicitly geometry-aware. The propagation of information between nodes is modulated by the local geometries, allowing the model to adapt its aggregation strategy based on the learned structure.

Geodesic Distance and Principal Curvatures Under the learned metric $\mathbf{G}_i$, the geodesic distance (in this locally constant metric space) between the feature vectors of two nodes i and j simplifies to a weighted Euclidean distance:

$$\begin{aligned} d_{\mathbf{G}_i}(\mathbf{h}_i, \mathbf{h}_j) &= \sqrt{(\mathbf{h}_i - \mathbf{h}_j)^T \mathbf{G}_i (\mathbf{h}_i - \mathbf{h}_j)} \\ &= \sqrt{\sum_{k=1}^{d} g_{i,k}(h_{i,k} - h_{j,k})^2} \end{aligned} \tag{3}$$

A major benefit of the diagonal form is that the principal directions of the geometry are aligned with the standard basis vectors $\{\mathbf{e}_k\}_{k=1}^d$, and the principal curvatures are directly related to the inverses of the metric elements $\{1/g_{i,k}\}_{k=1}^d$. This allows us to define a geometric modulation factor that captures how the geometry curves along the direction of the message.

Geometric Modulation and Attention We introduce a geometric modulation coefficient τ_{ij} that weights the message from node j to i based on the projection of their directional vector onto the principal axes, modulated by the local curvature in those directions. Let $\mathbf{d}_{ij} = (\mathbf{h}_j - \mathbf{h}_i)/\|\mathbf{h}_j - \mathbf{h}_i\|_2$ be the normalized direction vector. We define τ_{ij} as:

$$\tau_{ij} = \sum_{k=1}^{d} d_{ij,k}^2 \cdot \tanh(-\log g_{i,k}) \tag{4}$$

Here, the term $\tanh(-\log g_{i,k})$ acts as a curvature-dependent switch. If $g_{i,k}$ is large (high curvature, space con-

tracts), the term approaches -1; if $g_{i,k}$ is small (low curvature, space expands), it approaches $+1$.

Complementing this, we introduce a geometric attention mechanism α_{ij} that computes the similarity between nodes within their respective local geometries:

$$\alpha_{ij} = \frac{\langle \mathbf{h}_i, \mathbf{h}_j \rangle_{\mathbf{G}_i}}{\|\mathbf{h}_i\|_{\mathbf{G}_i} \|\mathbf{h}_j\|_{\mathbf{G}_j}} = \frac{\sum_k g_{i,k} h_{i,k} h_{j,k}}{\sqrt{\sum_k g_{i,k} h_{i,k}^2} \sqrt{\sum_k g_{j,k} h_{j,k}^2}} \tag{5}$$

This formulation measures the cosine similarity where the inner product is defined by node i's metric, and each node's norm is measured in its own local metric.

Node Representation Update The final message $\mathbf{m}_{ij}$ from node j to i integrates the geometric modulation and attention with a standard learnable transformation $\mathbf{W}_m$:

$$\mathbf{m}_{ij} = \tau_{ij} \cdot \sigma(\alpha_{ij}) \cdot \mathbf{W}_m \mathbf{h}_j \tag{6}$$

The node representations are then updated by aggregating messages from neighbors and combining them with the transformed self-representation, followed by a non-linear activation σ:

$$\mathbf{h}_i^{(l+1)} = \sigma\big(\mathbf{W}_s \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\big) \tag{7}$$

4.3 Ricci Flow-Inspired Geometric Regularization

To ensure that the learned metric field $\{\mathbf{G}_i\}$ is well-behaved and varies smoothly across the graph, we introduce two regularization terms inspired by the principles of discrete Ricci flow. This geometric evolution stabilizes training and prevents pathological curvatures.

First, we define a discrete approximation of the Ricci curvature for our diagonal metric field. Along the k-th principal direction at node i, let Ric_{kk} be the abbreviation of $\mathrm{Ric}(\mathbf{G}_i)_{kk}$, it can be approximated as:

$$\mathrm{Ric}_{kk}^{(i)} = \frac{1}{2|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \frac{g_{i,k} - g_{j,k}}{d_{\text{graph}}(i,j)}. \tag{8}$$

where $d_{\text{graph}}(i,j)$ is the shortest path distance on the graph, and node $j \in \mathcal{N}(i)$ makes $d_{\text{graph}} = 1$. See the detailed discussion in Appendix B.3. The **Ricci regularization** $\mathcal{L}_{\text{Ricci}}$ encourages the learned geometry to be Ricci-flat by penalizing the squared sum of Ricci curvatures, promoting a uniform curvature distribution:

$$\mathcal{L}_{\text{Ricci}} = \sum_{i \in \mathcal{V}} \sum_{k=1}^{d} (\mathrm{Ric}_{kk}^{(i)})^2 \tag{9}$$

The **smoothness regularization** $\mathcal{L}_{\text{smooth}}$ penalizes large differences in the metric vectors of adjacent nodes, ensuring the geometry varies smoothly over the graph structure:

$$\mathcal{L}_{\text{smooth}} = \sum_{(i,j) \in \mathcal{E}} \|\mathbf{g}_i - \mathbf{g}_j\|_2^2 \tag{10}$$

These regularizers guide the model to learn a coherent and stable geometric manifold that supports the downstream task. The total loss function then combines the primary task loss $\mathcal{L}_{\text{task}}$ with hyperparameters α, β:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{Ricci}} + \beta \mathcal{L}_{\text{smooth}} \tag{11}$$

5 Theoretical Analysis

We establish theoretical foundations for ARGNN, proving its convergence, stability analysis, and universality. All detailed proofs are provided in Appendix C.

5.1 Convergence and Stability Analysis

Theorem 1 (Convergence of Adaptive Geometry Learning)**.** *Consider ARGNN with L layers, hidden dimension d, and regularization weights α, β. Under mild regularity conditions, the learned metric tensors $\{\mathbf{G}_i\}_{i \in \mathcal{V}}$ converge to a stationary point with:*

$$\mathbb{E}\big[\|\nabla \mathcal{L}_{\text{total}}^{(t)}\|^2\big] = O\Big(\tfrac{1}{\sqrt{t}}\, e^{-\mu_{\text{eff}} t / L}\Big), \tag{12}$$

where μ_{eff} is the effective curvature of the loss landscape. Optimal regularization hyperparameters satisfy:

$$\alpha^\star = \Theta\big(\tfrac{\mathcal{H}}{L} \min\big(1, \tfrac{d}{|\mathcal{E}|}\big)\big), \quad \beta^\star = \Theta\big(\tfrac{\mathcal{H}\sqrt{d}}{|\mathcal{V}|}\big). \tag{13}$$

where $\mathcal{H} \in (0, 1]$ is the dataset-dependent homophily ratio.

Assume the theoretical optimal hyperparameters as

$$\alpha = \tfrac{c_1}{L} \cdot \min\big(1, \tfrac{d}{|\mathcal{E}|}\big), \quad \beta = \tfrac{c_2 \sqrt{d}}{|\mathcal{V}|} \tag{14}$$

For constants c_1, c_2, we get an effective practice as follows,

Proposition 1 (Homophily-Aware Constants)**.** *The dataset-dependent constants c_1, c_2 can be estimated as:*

$$c_1 \approx (1 - \mathcal{H}) + 0.1, \quad c_2 \approx 0.1 \cdot (1 + \mathcal{H}) \tag{15}$$

where $\mathcal{H} \in (0, 1]$ is the graph homophily ratio.

The proof (Appendix C.1) reveals key insights: i) Deeper networks require smaller α to maintain stability; ii) The smoothness weight β should scale with feature dimension and inversely with graph size; iii) The interplay between geometric regularization and task loss creates a favorable optimization landscape. These theoretical guidelines directly inform our hyperparameter choices in experiments.

5.2 Universal Approximation

Theorem 2 (Universal Geometric Framework)**.** *ARGNN provides a universal geometric framework that can generalize existing curvature-based GNNs. Specifically, GNNs operating on a fixed-curvature space (Euclidean, hyperbolic, spherical, or product manifolds) can be sufficiently approximated by a constrained parameterization of our learnable diagonal metric tensors $\mathbf{G}_i = diag(\mathbf{g}_i)$.*

See Appendix C.2 for the complete proof. We establish this by showing that fixed-curvature geometries correspond to specific constraints on $\mathbf{g}_i$: Euclidean ($\mathbf{g}_i = \mathbf{1}$), constant-curvature ($\mathbf{g}_i = c\mathbf{1}, c > 0$), and product manifolds (block-constant $\mathbf{g}_i$).

5.3 Computational Efficiency

Proposition 2 (Complexity Analysis)**.** *ARGNN has time complexity $O((n + m)d^2)$ per layer, where $n = |\mathcal{V}|$, $m = |\mathcal{E}|$, and d is the hidden feature dimension, matching standard GNNs while providing continuous geometric adaptation.*

See detailed proof in Appendix C.3. We also give the empirical comparison and analysis in Appendix E.

Dataset	Cora	CiteSeer	PubMed	Actor	Chameleon	Squirrel	Texas	Cornell	Wisconsin
Nodes	2,708	3,327	19,717	7,600	2,277	5,201	183	183	251
Edges	5,278	4,552	44,324	26,659	31,371	198,353	279	277	466
Features	1,433	3,703	500	932	2,325	2,089	1,703	1,703	1,703
Classes	7	6	3	5	5	5	5	5	5
$\mathcal{H}$	0.825	0.718	0.792	0.215	0.247	0.217	0.057	0.301	0.196

Table 1: Dataset Statistics and Homophily Ratios $\mathcal{H}$

6 Experiments

6.1 Experimental Setup

Datasets: We evaluate on nine widely used benchmark datasets spanning homophilic and heterophilic graphs: **Homophilic**: Cora, CiteSeer, PubMed (Sen et al. 2008); **Heterophilic**: Actor (Tang et al. 2009), Chameleon, Squirrel, Texas, Cornell, Wisconsin (Pei et al. 2020). Table 1 provides detailed statistics.

Baselines: We compare against four categories of methods: i) **Traditional GNNs**: GCN (Kipf and Welling 2016), GAT (Veličković et al. 2018), GraphSAGE (Hamilton, Ying, and Leskovec 2017); ii) **Geometric GNNs**: HGCN (Chami et al. 2019), HGAT (Zhang et al. 2021), κ-GCN (Bachmann, Bécigneul, and Ganea 2020), $\mathcal{Q}$-GCN (Xiong et al. 2022); iii) **Heterophilic GNNs**: H2GCN (Zhu et al. 2020), GPRGNN (Chien et al. 2020), FAGCN (Bo et al. 2021); iv) **Recent Methods**: CUSP (Grover et al. 2025), GNRF (Chen et al. 2025), CurvDrop (+JKNet) (Liu et al. 2023).

Implementation Details: We implemented on PyG (Fey and Lenssen 2019) and Geoopt (Kochurov, Karimov, and Kozlukov 2020) frameworks. All experiments were run on NVIDIA GPUs with 10 random seeds, with the given $60\%/20\%/20\%$ splits from (Pei et al. 2020) for the node classification task. And using the same $80\%/5\%/15\%$ splits and settings from (He et al. 2024) for the link existence prediction task.

6.2 Main Results

Table 2 presents mean F1-scores with 95% Confidence Intervals (CI) from 10 runs, chosen as the most comprehensive metric for multi-class imbalanced datasets. Additional metrics (Accuracy, AUROC, and AUPRC) are provided in Appendix E. ARGNN achieves the best performance on all 9 datasets, with significant improvements across both homophilic and heterophilic graphs. For instance, on Cora (homophilic), we improve 3.38% over CUSP, while on Wisconsin (heterophilic), we gain 2.35% over the best baseline.

Our results reveal three key patterns: i) Fixed-curvature methods like HGCN show high variance on heterophilic graphs, but generally overperform Euclidean-space GNNs; ii) Mixed-curvature approaches (CUSP, GNRF) achieve more consistent results but remain limited by discrete geometry choices; iii) ARGNN's continuous adaptation provides robust performance across the homophily spectrum.

For link prediction (Table 7 in Appendix), we report average AUROC as the comprehensive metric for link prediction. ARGNN achieves the highest scores on all 9 datasets, with particularly strong performance on geometrically complex graphs (Actor: 76.40% vs. GNRF's 73.50% and CUSP's 74.20%), validating our adaptive metric learning approach.

6.3 Ablation Studies

We conduct systematic ablations to validate our theoretical framework and understand the contribution of each component. Figure 3 presents results on three representative datasets: Cora (homophilic, $\mathcal{H} = 0.825$), Actor (heterophilic, $\mathcal{H} = 0.215$), and Wisconsin (mixed, $\mathcal{H} = 0.196$).

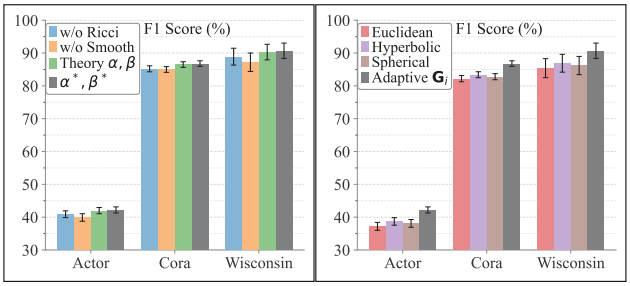

Figure 3: Ablation studies on three datasets. **(a)** Impact of regularization components, including theoretically optimal α, β settings v.s. grid search optimal α^*, β^* **(b)** Comparison with fixed $\mathbf{G}_i \in \{\mathbf{I}, 0.5\mathbf{I}, 2\mathbf{I}\}$ for Euclidean, Hyperbolic and Spherical. Error bars show 95% CI from 10 runs.

Regularization Components (Fig. 3a): Both Ricci and smoothness regularizations are essential but serve different roles. Removing Ricci regularization causes larger performance drops on heterophilic graphs (Actor: -1.3%, Wisconsin: -1.8%), where diverse geometries require careful control. Smoothness regularization is more critical for maintaining coherent metric fields, with Wisconsin showing the largest drop (-3.5%) due to its mixed homophily structure requiring smooth transitions between geometric regimes.

Remarkably, our theory-guided hyperparameters achieve near-optimal performance. Using theoretical optimal $\alpha = \frac{c_1}{L} \cdot \min(1, \frac{d}{|\mathcal{E}|})$ and $\beta = \frac{c_2\sqrt{d}}{|\mathcal{V}|}$ with c_1, c_2 from Theorem 1, we obtain F1 scores within 0.5% of exhaustive grid search. This validates our convergence analysis and provides practitioners with principled hyperparameter selection.

Geometry Parameterizations (Fig. 3b): Fixed geometries severely limit expressiveness across all graph types. Euclidean geometry ($\mathbf{G}_i = \mathbf{I}$), optimal for neither hierarchies nor cycles, shows the worst performance. Fixed hyperbolic geometry ($\mathbf{G}_i = 0.5\mathbf{I}$) improves slightly on tree-

Method	Homophilic			Heterophilic					
	Cora	CiteSeer	PubMed	Actor	Chameleon	Squirrel	Texas	Cornell	Wisconsin
GCN	75.21±0.28	67.30±1.05	83.75±0.07	31.12±0.96	61.16±0.23	43.06±0.33	75.61±0.07	67.72±1.19	59.46±3.25
GAT	76.70±0.13	66.23±0.85	82.83±0.22	32.65±0.23	63.10±0.77	43.90±0.01	76.09±0.77	74.01±0.01	55.29±5.40
GraphSAGE	71.88±0.91	70.01±0.64	81.09±0.13	36.73±0.01	59.99±0.89	41.11±1.16	77.11±0.45	69.91±0.24	81.18±3.45
HGCN	78.50±0.14	69.55±0.39	83.72±0.21	35.89±0.29	60.18±0.57	39.93±0.35	88.11±1.12	72.88±1.15	86.70±3.70
HGAT	77.12±0.01	70.12±0.92	84.02±0.19	35.12±0.27	62.43±0.59	41.78±0.37	85.56±1.10	73.12±0.18	87.20±3.50
κ-GCN	78.71±1.37	68.14±0.34	85.18±0.52	34.57±0.26	62.12±0.49	43.04±0.31	85.03±0.63	86.36±0.64	86.90±3.80
$\mathcal{Q}$-GCN	79.64±0.38	71.15±1.11	84.76±0.13	32.24±0.65	61.83±1.01	46.65±0.90	82.76±0.07	83.90±0.71	86.50±4.10
H2GCN	82.70±0.90	71.10±0.80	84.60±0.50	35.90±1.20	60.10±2.20	48.20±2.00	84.90±6.00	82.20±4.20	86.67±2.91
GPRGNN	79.49±0.31	67.61±0.38	84.07±0.09	37.43±1.09	65.09±0.43	47.51±0.23	88.34±0.09	87.21±0.70	88.07±1.00
FAGCN	82.50±0.50	71.30±0.60	84.30±0.40	35.80±1.10	66.90±1.80	52.30±1.70	87.80±3.20	85.50±4.10	87.30±3.60
CUSP	83.45±0.15	74.21±0.02	87.99±0.45	41.91±0.11	70.23±0.61	52.98±0.25	89.43±2.72	88.31±1.09	88.30±0.80
GNRF	82.10±0.80	73.50±0.50	86.80±0.40	40.80±0.90	68.90±1.20	49.82±1.50	90.80±1.30	90.50±1.10	88.10±1.40
CurvDrop	82.50±0.70	72.80±0.60	85.20±0.50	39.50±1.00	67.30±1.40	50.10±1.30	88.20±2.10	89.80±1.80	87.50±1.90
ARGNN	**86.83±0.84**	**74.80±1.26**	**88.59±0.25**	**42.18±0.33**	**70.44±1.27**	**53.12±1.45**	**92.28±1.59**	**90.85±0.33**	**90.65±2.34**

Table 2: Node Classification Performance (Avg. F1-score %(↑) ± 95% Confidence Interval) on Benchmark Datasets. The **bold**, purple and orange numbers denote the best, second best, and third best performances, respectively.

like substructures but fails on dense regions. Around 5% improvement of ARGNN over fixed geometries on heterophilic graphs demonstrates the necessity of adaptive metrics.

The performance gap is most pronounced on Actor (5.0% over Euclidean), where the co-occurrence network contains both star-like patterns (requiring hyperbolic geometry) and cliques (requiring spherical geometry), and our adaptive approach can simultaneously capture these diverse structures. Due to the space limit, detailed ablation results like additional parameterization comparisons and detailed computational overhead analysis are provided in Appendix E.

Computational Efficiency: ARGNN's diagonal parameterization achieves superior efficiency (nearly to HGCN): around 35% faster than CUSP by avoiding costly product manifold projections, with around 40% lower memory usage than full tensor methods. The $O(d)$ complexity per metric operation (v.s. $O(d^2)$ for full tensors) enables scaling to large graphs. Detailed benchmarks and scalability analysis are provided in Appendix E.

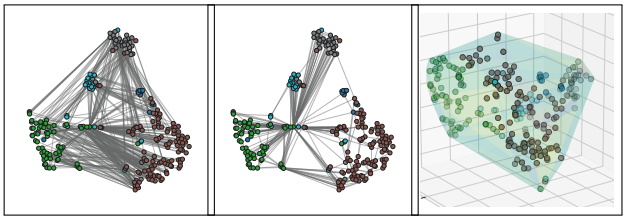

Figure 4: Geometry learned by ARGNN on Wisconsin. **Left**: Original graph topology colored by class under the layout from the learned embedding projection to 2-D. **Middle**: Degree-preserving rewiring based on learned geodesic distances reveals clearer class separation. **Right**: 3-D t-SNE embedding with curvature visualization shows adaptive geometry, the translucent hull is coloured by the magnitude of the mean curvature (violet→flat, yellow→strongly curved)

6.4 Learned Geometry Analysis

Figure 4 visualizes how ARGNN discovers latent geometric structure. The geodesic rewiring demonstrates improved class separation through learned metrics.

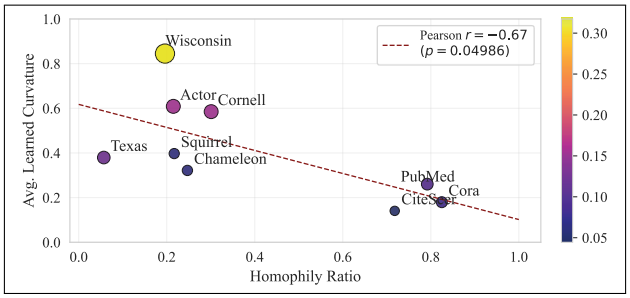

Figure 5: **Homophily $\mathcal{H}$ vs. learned geometry.** Avg. learned curvature across datasets with marker size/colour encodes the mean *Neighbour-Relative Metric Dispersion* (NRMD)

Figure 5 reveals that highly homophilic graphs (Cora, CiteSeer) are in the low curvature and *Neighbour-Relative Metric Dispersion* (NRMD $= \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} \frac{\|\mathbf{g}_i-\mathbf{g}_j\|_2}{\frac{1}{2}(\|\mathbf{g}_i\|_2+\|\mathbf{g}_j\|_2)}$) quadrant, whereas heterophilic datasets (Actor, Wisconsin) display both larger curvature and greater metric dispersion. The joint trend confirms that ARGNN bends its geometry more and allows higher learned metric diversity when neighbourhood labels are mixed, underscoring its dataset-adaptive behaviour. Due to the space limitation, see Appendix E for more evidence and analysis.

7 Conclusion

We have introduced ARGNN, a novel framework that learns continuous, anisotropic metric tensor fields to capture the geometric diversity inherent in real-world graphs.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No.62376236.

References

Bachmann, G.; Bécigneul, G.; and Ganea, O. 2020. Constant curvature graph convolutional networks. In *International conference on machine learning*, 486–496. PMLR.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3950–3957.

Chami, I.; Ying, Z.; Ré, C.; and Leskovec, J. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32.

Chen, J.; Deng, B.; Chen, C.; Zheng, Z.; et al. 2025. Graph neural ricci flow: Evolving feature from a curvature perspective. In *The Thirteenth International Conference on Learning Representations*.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2020. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*.

Chow, B.; and Knopf, D. 2004. *The ricci flow: An introduction: An introduction*, volume 1. American Mathematical Soc.

Cohen, T. S.; Geiger, M.; Köhler, J.; and Welling, M. 2018. Spherical CNNs. In *International Conference on Learning Representations*.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to construct knowledge bases from the World Wide Web. *Artificial intelligence*, 118(1-2): 69–113.

Fan, J. 2025a. Graph Minimum Factor Distance and Its Application to Large-Scale Graph Data Clustering. In *Forty-second International Conference on Machine Learning*.

Fan, J. 2025b. An Interdisciplinary and Cross-Task Review on Missing Data Imputation. *arXiv preprint arXiv:2511.01196*.

Fesser, L.; and Weber, M. 2024. Mitigating over-smoothing and over-squashing using augmentations of forman-ricci curvature. In *Learning on Graphs Conference*, 19–1. PMLR.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Forman. 2003. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete & Computational Geometry*, 29(3): 323–374.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. Pmlr.

Grover, K.; Yu, H.; Song, X.; Zhu, Q.; Xie, H.; Ioannidis, V. N.; and Faloutsos, C. 2025. Spectro-Riemannian Graph Neural Networks. *International Conference on Learning Representations*.

Gu, A.; Sala, F.; Gunel, B.; and Ré, C. 2018. Learning mixed-curvature representations in product spaces. In *International conference on learning representations*.

Gulcehre, C.; Denil, M.; Malinowski, M.; Razavi, A.; Pascanu, R.; Hermann, K. M.; Battaglia, P.; Bapst, V.; Raposo, D.; Santoro, A.; et al. 2018. Hyperbolic attention networks. In *International Conference on Learning Representations*.

Guo, Z.; Sun, Q.; Yuan, H.; Fu, X.; Zhou, M.; Gao, Y.; and Li, J. 2025. GraphMoRE: Mitigating Topological Heterogeneity via Mixture of Riemannian Experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11754–11762.

Hamilton, R. S. 1982. Three-manifolds with positive Ricci curvature. *Journal of Differential geometry*, 17(2): 255–306.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, Y.; Zhang, X.; Huang, J.; Rozemberczki, B.; Cucuringu, M.; and Reinert, G. 2024. Pytorch geometric signed directed: a software package on graph neural networks for signed and directed graphs. In *Learning on Graphs Conference*, 12–1. PMLR.

Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; and Tang, J. 2020. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*.

Kang, Q.; Zhao, K.; Song, Y.; Wang, S.; and Tay, W. P. 2023. Node embedding from neural Hamiltonian orbits in graph neural networks. In *International conference on machine learning*, 15786–15808. PMLR.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kochurov, M.; Karimov, R.; and Kozlukov, S. 2020. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*.

Lee, J. M. 2018. *Introduction to Riemannian manifolds*, volume 2. Springer.

Li, J.; Fu, X.; Sun, Q.; Ji, C.; Tan, J.; Wu, J.; and Peng, H. 2022. Curvature graph generative adversarial networks. In *Proceedings of the ACM web conference 2022*, 1528–1537.

Liu, Y.; Zhou, C.; Pan, S.; Wu, J.; Li, Z.; Chen, H.; and Zhang, P. 2023. Curvdrop: A ricci curvature based approach to prevent graph neural networks from over-smoothing and over-squashing. In *Proceedings of the ACM Web Conference 2023*, 221–230.

Ni, C.-C.; Lin, Y.-Y.; Luo, F.; and Gao, J. 2019. Community detection on networks with Ricci flow. *Scientific reports*, 9(1): 9984.

Obata, M. 1970. Conformal transformations of Riemannian manifolds. *Journal of Differential Geometry*, 4(3): 311–333.

Ollivier, Y. 2009. Ricci curvature of Markov chains on metric spaces. *Journal of Functional Analysis*, 256(3): 810–864.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*.

Qian, F.; Bai, L.; Cui, L.; Li, M.; Lyu, Z.; Du, H.; and Hancock, E. 2025. DHAKR: Learning Deep Hierarchical Attention-Based Kernelized Representations for Graph Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 19995–20003.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shen, C.; Ding, P.; Wee, J.; Bi, J.; Luo, J.; and Xia, K. 2024. Curvature-enhanced graph convolutional network for biomolecular interaction prediction. *Computational and Structural Biotechnology Journal*, 23: 1016–1025.

Spivak, M. 1999. *A comprehensive introduction to differential geometry*, volume 2. Publish or Perish.

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

Sun, Y.; and Fan, J. 2024. MMD Graph Kernel: Effective Metric Learning for Graphs via Maximum Mean Discrepancy. In *The Twelfth International Conference on Learning Representations*.

Sun, Z.; Ding, C.; and Fan, J. 2023. Lovász Principle for Unsupervised Graph Representation Learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 58290–58311. Curran Associates, Inc.

Sun, Z.; Wang, X.; Ding, C.; and Fan, J. 2024. Learning Graph Representation via Graph Entropy Maximization. In *International Conference on Machine Learning*, 47133–47158. PMLR.

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

Topping, J.; Di Giovanni, F.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. M. 2021. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, X.; Ding, C.; Li, T.; and Fan, J. 2025a. Adaptive riemannian graph neural networks. *arXiv preprint arXiv:2508.02600*.

Wang, X.; Sun, Z.; Ding, C.; and Fan, J. 2025b. Explainable Graph Representation Learning via Graph Pattern Analysis. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 3426–3434. International Joint Conferences on Artificial Intelligence Organization.

Wang, X.; Sun, Z.; Ding, C.; and Fan, J. 2025c. Learnable Kernel Density Estimation for Graphs. *arXiv preprint arXiv:2505.21285*.

Wang, Z.; and Fan, J. 2024. Graph classification via reference distribution learning: theory and practice. *Advances in Neural Information Processing Systems*, 37: 137698–137740.

Wu, J.; Chen, H.; Cheng, M.; and Xiong, H. 2023. Curvagn: curvature-based adaptive graph neural networks for predicting protein-ligand binding affinity. *BMC bioinformatics*, 24(1): 378.

Xiong, B.; Zhu, S.; Potyka, N.; Pan, S.; Zhou, C.; and Staab, S. 2022. Pseudo-riemannian graph convolutional networks. *Advances in Neural Information Processing Systems*, 35: 3488–3501.

Zhang, Y.; Wang, X.; Shi, C.; Jiang, X.; and Ye, Y. 2021. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 8(6): 1690–1701.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33: 7793–7804.