R-MESD: RIEMANNIAN MULTI EXPERT LEARNING FOR STRUCTURAL-SEMANTIC DECOUPLING IN TEXT-ATTRIBUTED GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation learning on text-attributed graphs (TAGs) is crucial for modeling rich node semantics and complex graph structure. Nevertheless, this task faces structural-semantic mismatch arising from divergent modality distributions, as well as heterogeneous local topologies that a single geometric space cannot adequately capture. Existing approaches typically propagate semantic features along the given topology or rely on a fixed representation space, potentially amplifying modality interference and overlooking local geometric diversity. To address these issues, we propose R-MESD, a Riemannian Multi Expert Structural-Semantic Decoupling model in TAGs. Specifically, R-MESD initializes structural positions from Laplacian spectral signals and uses local structural descriptions to route nodes to Hyperbolic and Spherical experts. The expert representations from different manifolds are transformed into a shared latent space within each geometric branch and subsequently fused according to node-level routing weights, followed by cross-geometry attention. Meanwhile, node semantics are represented as tangent vectors anchored at the learned structural positions and further refined through cross-geometry semantic alignment and local tangent-space attention. Extensive experiments on five public benchmarks demonstrate that R-MESD consistently outperforms state-of-the-art baselines.

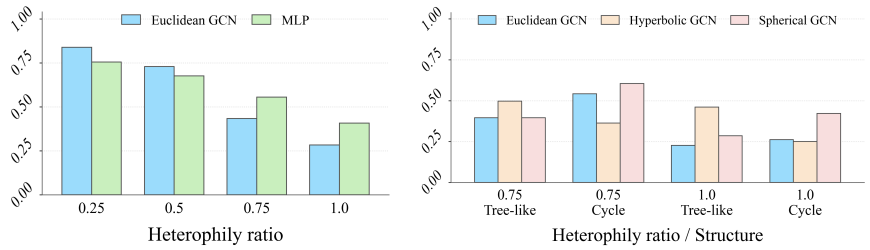
1 INTRODUCTION

Text-attributed graphs (TAGs) associate entities with textual descriptions and represent their relationships through graph topology, providing a natural representation of academic networks, e-commerce platforms, and social media (Yang et al., 2015; Yan et al., 2023). Unlike plain graphs that focus exclusively on connectivity, TAGs offer both rich node semantics and structural information, whose joint modeling is essential for learning informative node representations (Yang et al., 2021; Wang et al., 2024; Zhou et al., 2025a). Graph neural networks (GNNs) typically integrate these sources by propagating and aggregating node attributes along graph topology (Kipf & Welling, 2017; Velickovic et al., 2018; Hamilton et al., 2017). However, such propagation relies on the consistency between node semantics and local connectivity. When these sources disagree, neighborhood aggregation may introduce irrelevant or conflicting information and amplify interference across layers. We refer to this discrepancy as **structural-semantic mismatch**, where semantic relations expressed by node text are inconsistent with local connectivity patterns (Zhou et al., 2025b).

To provide further insight into this issue, we take the ArXiv citation graph as an example and examine the effect of structural-semantic mismatch on neighborhood propagation. We quantify the mismatch using the heterophily ratio (Zhu et al., 2020; Pei et al., 2020; Li et al., 2025), divide test nodes into four heterophily intervals, and compare a Euclidean graph convolutional network (GCN) with a multilayer perceptron (MLP) using only node semantics. As shown in Figure 1(a), GCN outperforms MLP in the two lower-heterophily intervals but falls behind in the two higher ones. This result indicates that fixed neighborhood propagation is not universally beneficial and can introduce additional interference as structural-semantic mismatch becomes more pronounced.

This observation raises a further question: can an alternative geometric space better accommodate the diverse local structures under high structural-semantic mismatch? To this end, we compare Eu-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Propagation reliability *(b)* Local geometric preferences

Figure 1: Motivation results (Accuracy). (a) Euclidean GCN and semantic-only MLP across heterophily intervals. (b) Euclidean, Hyperbolic, and Spherical GCNs in Tree-like and Cycle groups.

clidean, Hyperbolic, and Spherical GCNs on Tree-like and Cycle groups within the same graph, focusing on the two higher-heterophily intervals. As shown in Figure 1(b), Hyperbolic GCN performs best on Tree-like groups, whereas Spherical GCN achieves higher accuracy in Cycle groups (Chami et al., 2019; Igashov et al., 2021). Both outperform Euclidean GCN in their respective groups. These results suggest that, under high structural-semantic mismatch, the effectiveness of neighborhood propagation depends not only on the degree of mismatch but also on the compatibility between the underlying geometry and local topology. This motivates adaptive multi-geometry representation learning that can accommodate heterogeneous structural characteristics.

Taken together, these observations reveal two limitations of fixed propagation. First, aggregating node attributes along the given topology can introduce interference in high-heterophily neighborhoods, potentially obscuring node-specific semantics. Second, local structures exhibit different geometric preferences, making a single propagation space inadequate for diverse structural patterns. More importantly, changing the propagation geometry alone does not explicitly separate structural positions from semantic attributes; jointly encoding and propagating them may still allow structural aggregation to interfere with node-specific semantics. The key challenge is therefore to mitigate the interference of structural propagation with node semantics while capturing the diverse geometric characteristics of local structures.

To address the aforementioned issues, we propose a novel **R**iemannian **M**ulti **E**xpert **S**tructural-**S**emantic **D**ecoupling model for representation learning on TAGs, termed R-MESD. Specifically, R-MESD assigns distinct representational roles to structure and semantics, using manifold positions for topology and tangent vectors anchored at these positions for node semantics. To capture diverse local structures, the structural branch starts from topology-based initialization and uses local structural descriptions to adaptively weight Hyperbolic and Spherical experts at each node. Subsequently, geometric transformations support weighted fusion within each geometric branch, while cross-geometry attention exchanges structural information between the two views. On the semantic side, cross-geometry alignment enables information exchange between geometric views, followed by local tangent-space attention that aggregates neighborhood semantics under the guidance of the learned structural positions. In this way, R-MESD captures diverse local structures through multiple curvature experts while decoupling structural positions from semantic representations.
The main contributions are summarized as follows:

- We identify two limitations of most existing TAG representation learning methods, that is, fixed neighborhood propagation can introduce semantic interference under structural-semantic mismatch, while a single geometric space may not adequately capture diverse local structures.
- We propose R-MESD, a Riemannian multi-expert structural-semantic decoupling model that captures diverse local structures with Hyperbolic and Spherical experts, while decoupling and refining node semantics through tangent-space representations and geometric-view alignment.
- Experiments on five public benchmarks demonstrate substantial improvements over state-of-the-art baselines, supporting R-MESD's effectiveness.

2 PRELIMINARY

Notation and Problem Formulation. Given a text-attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ denotes the set of n nodes, and $\mathcal{E}$ represents the set of m edges connecting these nodes. For any edge $(v_i, v_j) \in \mathcal{E}$, nodes v_i and v_j are considered neighbors. We denote the set of one-hop

neighbors of node v_i as $\mathcal{N}(v_i) = \{v_j \mid (v_i, v_j) \in \mathcal{E}\}$. In a TAG, each node v_i is associated with a raw text description $t_i \in \mathcal{T}$, which encapsulates its semantic information. The primary objective is to learn node representations from graph structure and node semantics for downstream tasks (Yan et al., 2023; Su et al., 2025).

Riemannian Geometry. Riemannian geometry provides a systematic framework for structural analysis (Sun & Yu, 2026), with its elementary object being a Riemannian manifold, a smooth manifold $\mathcal{M}^d$ endowed with a Riemannian metric $\boldsymbol{g}$, where d denotes the dimensionality.

We formulate our model in the κ-stereographic model, which unifies Hyperbolic ($\kappa < 0$), Spherical ($\kappa > 0$), and Euclidean ($\kappa = 0$) geometries in a common coordinate system (Bachmann et al., 2020). Specifically, the d-dimensional space with curvature κ is given by $\mathcal{M}_\kappa^d = \left\{\boldsymbol{x} \in \mathbb{R}^d : -\kappa\|\boldsymbol{x}\|_2^2 < 1\right\}$. Each point $\boldsymbol{x} \in \mathcal{M}_\kappa^d$ has a tangent space $T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$, whose vectors can be represented by coordinates in $\mathbb{R}^d$. The manifold carries a Riemannian metric $\boldsymbol{g}_{\boldsymbol{x}}^\kappa$ at each point $\boldsymbol{x}$; its conformal factor $\lambda_{\boldsymbol{x}}^\kappa$ determines the local scaling relative to the Euclidean metric. For tangent vectors $\boldsymbol{a}, \boldsymbol{b} \in T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$, $\langle \boldsymbol{a}, \boldsymbol{b}\rangle_{\boldsymbol{x}}^\kappa$ denotes the Riemannian inner product induced by $\boldsymbol{g}_{\boldsymbol{x}}^\kappa$.

For $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}_\kappa^d$, the logarithmic map at $\boldsymbol{x}$ is $\mathrm{Log}_{\boldsymbol{x}}^\kappa(\cdot) : \mathcal{M}_\kappa^d \to T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$. Conversely, the exponential map $\mathrm{Exp}_{\boldsymbol{x}}^\kappa$ maps a tangent vector back to the manifold. To connect tangent spaces at different points, parallel transport $\mathrm{PT}_{\boldsymbol{x}\to\boldsymbol{y}}^\kappa$ is a linear isometry that transports a vector $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$ to $T_{\boldsymbol{y}}\mathcal{M}_\kappa^d$ along the geodesic connecting $\boldsymbol{x}$ and $\boldsymbol{y}$ on the manifold.

We use $\boldsymbol{o} = \boldsymbol{0}$ as the coordinate origin. Scalar multiplication is defined by $r \otimes_\kappa \boldsymbol{x} = \mathrm{Exp}_{\boldsymbol{o}}^\kappa(r\,\mathrm{Log}_{\boldsymbol{o}}^\kappa(\boldsymbol{x}))$ for $r \in \mathbb{R}$. To aggregate manifold-valued representations, we use the geometric centroid $m_\kappa(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \boldsymbol{\alpha})$, which combines manifold coordinates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ with nonnegative weights $\boldsymbol{\alpha}$. The corresponding coordinate expressions are collected in Appendix D.

3 METHODOLOGY

We start with an overview of the proposed method and then explain its key stages.

3.1 OVERVIEW

To address structural-semantic mismatch and heterogeneous local topologies in TAGs, we propose a Riemannian Multi Expert Structural-Semantic Decoupling model, termed R-MESD. The overall architecture is shown in Figure 2. Specifically, R-MESD employs Hyperbolic and Spherical experts to capture complementary geometric characteristics of graph topology, with curvature-aware routing assigning node-specific weights based on local structural descriptions. The resulting expert representations are transformed into shared reference manifolds within each geometric branch and fused through weighted centroids, followed by bidirectional cross-geometry structural attention to exchange information between the two geometric views. In parallel, text encoders map node texts to tangent vectors anchored at the learned structural positions. Cross-geometry semantic alignment then facilitates information exchange between the two views, while local tangent-space attention aggregates neighborhood semantics according to the learned structural positions. Consequently, R-MESD produces, for each node, a structural position and a semantic tangent vector in both Hyperbolic and Spherical views.

3.2 ADAPTIVE RIEMANNIAN STRUCTURAL EXPERTS

Considering the diverse local topologies of TAG neighborhoods and their potentially different geometric preferences, we employ Hyperbolic and Spherical experts, with local structural descriptions guiding their adaptive routing.

Structural Initialization. We initialize node positions based on graph topology. Formally, let $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ denote the normalized Laplacian, with $\boldsymbol{A}$ and $\boldsymbol{D}$ the adjacency and degree matrices. Let $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]$ collect the d eigenvectors of $\boldsymbol{L}$ with the largest eigenvalues. For node v_i, the corresponding row $\boldsymbol{s}_i = \boldsymbol{U}_{i,:}$ serves as its spectral representation. For $\kappa \neq 0$, we set $r_\kappa = \rho/\sqrt{|\kappa|}$ with a shared $\rho \in (0, 1)$ and initialize:

$$\boldsymbol{p}_i^{(0,\kappa)} = F_\kappa(\boldsymbol{s}_i) = r_\kappa \frac{\boldsymbol{s}_i}{\|\boldsymbol{s}_i\|_2 + \varepsilon}, \tag{1}$$

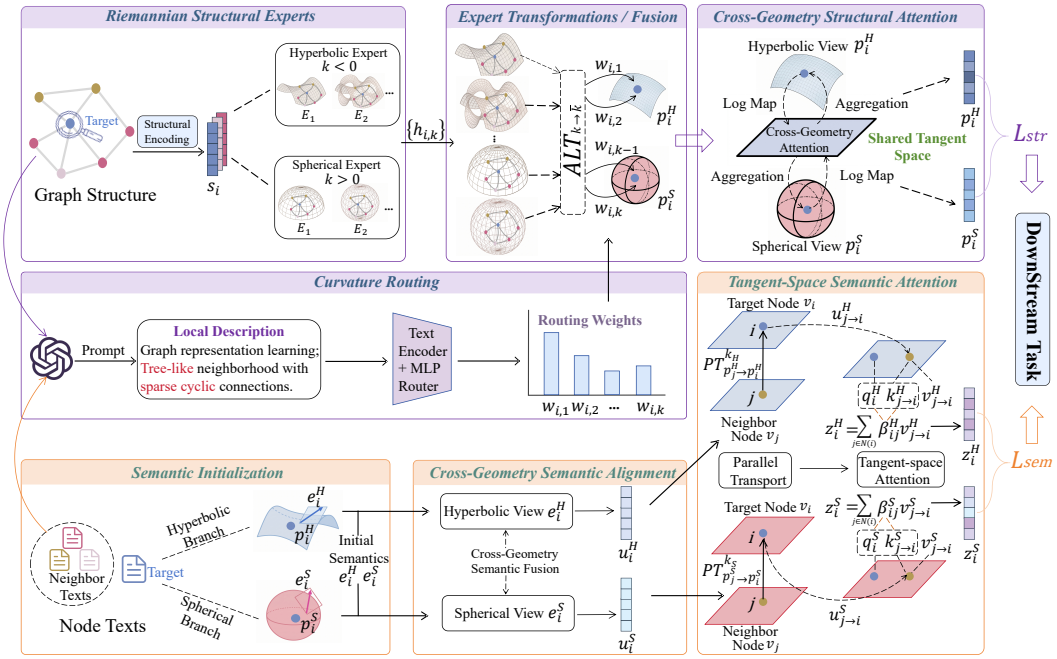


Figure 2: The whole architecture of our proposed R-MESD.

where $\varepsilon > 0$ stabilizes normalization, and ρ scales the initial positions based on the curvature radius.

Theorem 1 (Lipschitz-Stable Manifold Initialization) *For the curvature-adaptive map F_κ defined in Eq. 1, if $\varepsilon > 0$, $\kappa \neq 0$, and $r_\kappa = \rho/\sqrt{|\kappa|}$ with $\rho \in (0, 1)$, then for any $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^d$, the following two properties hold:*

$$F_\kappa(\boldsymbol{s}) \in \mathcal{M}_\kappa^d, \qquad \|F_\kappa(\boldsymbol{s}) - F_\kappa(\boldsymbol{t})\|_2 \leq \frac{r_\kappa}{\varepsilon}\|\boldsymbol{s} - \boldsymbol{t}\|_2. \tag{2}$$

The proof is provided in Appendix C.1.

Geometric Experts. To capture diverse local topologies, we employ Hyperbolic experts for tree-like and hierarchical patterns, while Spherical experts for cyclic and periodic patterns (Chami et al., 2019; Igashov et al., 2021). The corresponding curvature sets are defined as $\mathcal{K}_H \subset \mathbb{R}_{<0}$ and $\mathcal{K}_S \subset \mathbb{R}_{>0}$, respectively. For each expert $\kappa \in \mathcal{K}_H \cup \mathcal{K}_S$, we initialize the row associated with node v_i as $(\boldsymbol{H}_\kappa^{(0)})_{i,:} = \boldsymbol{p}_i^{(0,\kappa)}$. During learning, the expert preserves manifold-valued representations by first mapping them to the tangent space at the origin with $\mathrm{Log}_{\boldsymbol{o}}^\kappa$, applying an expert-specific linear transformation, and then mapping the transformed representations back to the manifold with:

$$\widetilde{\boldsymbol{H}}_\kappa^{(l)} = \mathrm{Exp}_{\boldsymbol{o}}^\kappa\left(\mathrm{Log}_{\boldsymbol{o}}^\kappa\left(\boldsymbol{H}_\kappa^{(l)}\right)\boldsymbol{W}_\kappa^{(l)}\right), \tag{3}$$

where $\boldsymbol{W}_\kappa^{(l)}$ is the expert-specific weight matrix. Based on this, neighborhood representations are aggregated through the geometric centroid with normalized adjacency weights $\widehat{\boldsymbol{A}}_{ij}$:

$$\boldsymbol{m}_{i,\kappa}^{(l)} = m_\kappa\left(\left\{\widetilde{\boldsymbol{h}}_{j,\kappa}^{(l)} : v_j \in \widetilde{\mathcal{N}}(v_i)\right\}; \left\{\widehat{\boldsymbol{A}}_{ij} : v_j \in \widetilde{\mathcal{N}}(v_i)\right\}\right), \tag{4}$$

where $\widetilde{\mathcal{N}}(v_i)$ denotes the one-hop neighborhood of node v_i together with v_i itself.

Then, we apply the nonlinearity σ in the tangent space and map the result back to the manifold:

$$\boldsymbol{h}_{i,\kappa}^{(l+1)} = \mathrm{Exp}_{\boldsymbol{o}}^\kappa\left(\sigma\left(\mathrm{Log}_{\boldsymbol{o}}^\kappa\left(\boldsymbol{m}_{i,\kappa}^{(l)}\right)\right)\right). \tag{5}$$

After L layers, each expert produces the structural representation $\boldsymbol{h}_{i,\kappa} = \boldsymbol{h}_{i,\kappa}^{(L)}$ for node v_i.

Curvature Routing. To adapt expert selection to each node's local topology, we construct routing evidence from local structural statistics and textual context. For each node v_i, the evidence is:

$$\mathcal{Q}_i = \left(d_i, t_i, \mathcal{T}_{\mathcal{N}(v_i)}, \rho_i^{\text{tree}}, \rho_i^{\text{cycle}}, \rho_i^{\text{hier}}\right), \tag{6}$$

where d_i denotes the degree of node v_i, the textual context comprises the node text t_i and the texts of its neighbors, $\mathcal{T}_{\mathcal{N}(v_i)}$. The local structural descriptors ρ_i^{tree}, ρ_i^{cycle}, and ρ_i^{hier} quantify the tree-like, cyclic, and hierarchical tendencies of the neighborhood, respectively.

Using this evidence, we construct a constrained prompt template $\mathcal{P}$ that guides a large language model (LLM) to generate a structured local description q_i, which is then encoded into $\boldsymbol{c}_i$ by a frozen sentence-level text encoder (Reimers & Gurevych, 2019):

$$q_i = \text{LLM}\left(\mathcal{P}(\mathcal{Q}_i)\right), \qquad \boldsymbol{c}_i = \text{Enc}_{\text{sent}}(q_i). \tag{7}$$

After that, an MLP with a softmax layer transforms $\boldsymbol{c}_i$ into node-level routing weights:

$$\boldsymbol{a}_i = \text{MLP}_{\text{route}}(\boldsymbol{c}_i), \qquad \boldsymbol{w}_i = \text{softmax}(\boldsymbol{a}_i), \tag{8}$$

where $w_{i,\kappa}$ denotes the weight assigned to expert κ for node v_i, and the weights satisfy $\sum_{\kappa \in \mathcal{K}_H \cup \mathcal{K}_S} w_{i,\kappa} = 1$. The routing weights subsequently determine the contribution of each expert to the geometric fusion.

3.3 Riemannian Transformations and Cross Geometry Attention

Expert outputs reside on manifolds with different curvatures, making direct fusion incompatible with their manifold constraints. To enable their integration, we apply geometric transformations to map the outputs within each branch to a shared reference manifold, where they are fused using the node-level routing weights. Bidirectional cross-geometry attention then enables structural information exchange between the Hyperbolic and Spherical views.

Expert Transformations and Fusion. Within each geometric branch, expert representations are defined on manifolds with different curvatures. To align them in a common geometric space, we introduce the Augmented Lorentz Transformation (ALT), which maps all expert representations within a branch to a shared reference manifold, that is, $\mathcal{H}^d$ for the Hyperbolic branch and $\mathcal{S}^d$ for the Spherical branch (Sun et al., 2025). The target curvature $\bar{\kappa}$ is determined by the selected branch. Specifically, the inverse stereographic projection Π_κ^{-1} lifts each representation to a unified Lorentz/sphere spacetime model, yielding:

$$x_t = \frac{\lambda^\kappa_{\boldsymbol{h}_{i,\kappa}} - 1}{\sqrt{|\kappa|}}, \qquad \boldsymbol{x}_s = \lambda^\kappa_{\boldsymbol{h}_{i,\kappa}} \boldsymbol{h}_{i,\kappa}. \tag{9}$$

Definition 1 (Unified Lorentz/Sphere Spacetime Model) *For $\kappa \neq 0$, the m-dimensional Lorentz/sphere spacetime model is defined as:*

$$\mathcal{L}_\kappa^m = \left\{ [x_t; \boldsymbol{x}_s] \in \mathbb{R}^{m+1} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle_\kappa = \frac{1}{\kappa} \right\}, \tag{10}$$

where the spacetime form is defined as follows:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_\kappa = \text{sgn}(\kappa) x_t y_t + \boldsymbol{x}_s^\top \boldsymbol{y}_s. \tag{11}$$

The lifted coordinate $\boldsymbol{x} \in \mathcal{L}_\kappa^{d_\kappa}$ satisfies this constraint. The ALT then aligns each representation's curvature and spatial dimensionality with the corresponding branch reference manifold:

$$\boldsymbol{y} = \text{ALT}_{\kappa \to \bar{\kappa}}(\boldsymbol{W}_\kappa, \boldsymbol{x}) = \begin{bmatrix} \gamma_\kappa(\boldsymbol{x}_s) & \boldsymbol{0}^\top \\ \boldsymbol{0} & \boldsymbol{W}_\kappa \end{bmatrix} \begin{bmatrix} x_t \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} \gamma_\kappa(\boldsymbol{x}_s) x_t \\ \boldsymbol{W}_\kappa \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} y_t \\ \boldsymbol{y}_s \end{bmatrix}, \tag{12}$$

where $\boldsymbol{W}_\kappa \in \mathbb{R}^{d \times d_\kappa}$ is a learnable spatial transformation.

After that, let $\ell_\kappa(\boldsymbol{x}_s) = \|\boldsymbol{W}_\kappa \boldsymbol{x}_s\|_2^2$, the temporal correction factor can be defined as:

$$\gamma_\kappa(\boldsymbol{x}_s) = \sqrt{\frac{|\kappa|}{|\bar{\kappa}|} \frac{1 - \bar{\kappa} \ell_\kappa(\boldsymbol{x}_s)}{1 - \kappa \|\boldsymbol{x}_s\|_2^2}}. \tag{13}$$

Finally, we project $\boldsymbol{y}$ onto the target manifold to obtain the transformed representation:

$$\widetilde{\boldsymbol{h}}_{i,\kappa} = \Pi_{\bar{\kappa}}(\boldsymbol{y}) = \frac{\boldsymbol{y}_s}{1 + \sqrt{|\bar{\kappa}|}\, y_t}. \tag{14}$$

Equivalently, the transformed representation can be expressed compactly as:

$$\widetilde{\boldsymbol{h}}_{i,\kappa} = \Pi_{\bar{\kappa}} \left(\mathrm{ALT}_{\kappa \to \bar{\kappa}} \left(\boldsymbol{W}_\kappa, \Pi_\kappa^{-1}(\boldsymbol{h}_{i,\kappa}) \right) \right). \tag{15}$$

Theorem 2 (Target Manifold Preservation under ALT) *For $\kappa, \bar{\kappa} \neq 0$, let $\boldsymbol{x} \in \mathcal{L}_\kappa^{d_\kappa}$ and define $\boldsymbol{y} = \mathrm{ALT}_{\kappa \to \bar{\kappa}}(\boldsymbol{W}_\kappa, \boldsymbol{x})$. If $\gamma_\kappa(\boldsymbol{x}_s)$ and the forward projection are well defined, then:*

$$\boldsymbol{y} \in \mathcal{L}_{\bar{\kappa}}^d, \qquad \Pi_{\bar{\kappa}}(\boldsymbol{y}) \in \mathcal{M}_{\bar{\kappa}}^d. \tag{16}$$

The proof is provided in Appendix C.2.

After aligning expert representations across curvatures and spatial dimensions, expert representations in each branch are fused on the corresponding reference manifold using routing weights:

$$\boldsymbol{p}_i^\xi = m_{\kappa_\xi} \left(\{\widetilde{\boldsymbol{h}}_{i,\kappa}\}_{\kappa \in \mathcal{K}_\xi}; \{w_{i,\kappa}\}_{\kappa \in \mathcal{K}_\xi} \right), \qquad \xi \in \{H, S\}. \tag{17}$$

This fusion yields separate Hyperbolic and Spherical structural positions, $\boldsymbol{p}_i^H \in \mathcal{H}^d$ and $\boldsymbol{p}_i^S \in \mathcal{S}^d$.

Cross-Geometry Structural Attention. To exchange structural information across the two geometric branches, we introduce bidirectional cross-geometry attention. In the Hyperbolic update, queries come from the Spherical branch, while keys and values come from the Hyperbolic branch. The corresponding branch-specific manifold linear maps are defined as:

$$\boldsymbol{q}_i^S = \boldsymbol{W}_Q^S \otimes_{\kappa_S} \boldsymbol{p}_i^S, \qquad \boldsymbol{k}_j^H = \boldsymbol{W}_K^H \otimes_{\kappa_H} \boldsymbol{p}_j^H, \qquad \boldsymbol{v}_j^H = \boldsymbol{W}_V^H \otimes_{\kappa_H} \boldsymbol{p}_j^H. \tag{18}$$

The resulting queries and keys are then mapped to the tangent spaces at the origin $\boldsymbol{o}$:

$$\overline{\boldsymbol{q}}_i^S = \mathrm{Log}_{\boldsymbol{o}}^{\kappa_S}(\boldsymbol{q}_i^S), \qquad \overline{\boldsymbol{k}}_j^H = \mathrm{Log}_{\boldsymbol{o}}^{\kappa_H}(\boldsymbol{k}_j^H). \tag{19}$$

The cross-geometry attention weights are computed at the origin $\boldsymbol{o}$ under curvature κ_H:

$$\alpha_{ij}^H = \frac{\exp\left(\left\langle \overline{\boldsymbol{q}}_i^S, \overline{\boldsymbol{k}}_j^H \right\rangle_{\boldsymbol{o}}^{\kappa_H} \right)}{\sum_{v_m \in \mathcal{V}} \exp\left(\left\langle \overline{\boldsymbol{q}}_i^S, \overline{\boldsymbol{k}}_m^H \right\rangle_{\boldsymbol{o}}^{\kappa_H} \right)}. \tag{20}$$

Using attention weights, the values are aggregated to update the Hyperbolic structural position $\boldsymbol{p}_i^H$:

$$\boldsymbol{p}_i^H = m_{\kappa_H} \left(\{\boldsymbol{v}_j^H\}_{v_j \in \mathcal{V}}; \{\alpha_{ij}^H\}_{v_j \in \mathcal{V}} \right). \tag{21}$$

The Spherical branch is updated analogously. This interaction integrates complementary structural information from both geometric branches.

3.4 Riemannian Semantic Alignment and Tangent Space Attention

To capture node-specific semantic information, we represent node semantics as tangent vectors anchored at the learned structural positions. On this basis, cross-geometry alignment and local tangent-space attention integrate semantic information across geometric branches and neighborhoods.

Semantic Initialization. For node v_i, two independently learnable mappings, ψ_H and ψ_S, encode its text t_i as semantic tangent vectors anchored at its Hyperbolic and Spherical structural positions:

$$\boldsymbol{e}_i^H = \psi_H(t_i) \in T_{\boldsymbol{p}_i^H} \mathcal{H}^d, \qquad \boldsymbol{e}_i^S = \psi_S(t_i) \in T_{\boldsymbol{p}_i^S} \mathcal{S}^d. \tag{22}$$

In the κ-stereographic model, these tangent vectors serve as the initial semantic representations for the two geometric branches.

Cross-Geometry Semantic Alignment. To exchange semantic information between geometric branches, we apply learnable linear maps in local tangent coordinates. Formally, for node v_i, the aligned Spherical representation is fused with its Hyperbolic counterpart, formulated as:

$$\boldsymbol{u}_i^H = \boldsymbol{W}_F^H \left[\boldsymbol{e}_i^H \,\|\, \boldsymbol{W}_{S\to H} \boldsymbol{e}_i^S \right] + \boldsymbol{b}_F^H, \qquad \boldsymbol{u}_i^H \in T_{\boldsymbol{p}_i^H} \mathcal{H}^d. \tag{23}$$

The Spherical branch is updated analogously. This fusion combines semantic information from both geometric branches.

Tangent-Space Semantic Attention. Neighboring semantic vectors lie in different tangent spaces. Therefore, we use parallel transport to map the fused semantic representation $\boldsymbol{u}_j^H$ to the target node's tangent space $T_{\boldsymbol{p}_i^H} \mathcal{H}^d$:

$$\boldsymbol{u}_{j\to i}^H = \mathrm{PT}^{\kappa_H}_{\boldsymbol{p}_j^H \to \boldsymbol{p}_i^H} \left(\boldsymbol{u}_j^H \right). \tag{24}$$

Within this tangent space, the target node's initial semantic representation provides the query, whereas the transported neighbor representations provide the keys and values:

$$\boldsymbol{q}_i^H = \boldsymbol{U}_Q^H \boldsymbol{e}_i^H, \qquad \boldsymbol{k}_{j\to i}^H = \boldsymbol{U}_K^H \boldsymbol{u}_{j\to i}^H, \qquad \boldsymbol{v}_{j\to i}^H = \boldsymbol{U}_V^H \boldsymbol{u}_{j\to i}^H. \tag{25}$$

Then, attention weights are computed at the learned structural position $\boldsymbol{p}_i^H$:

$$\beta_{ij}^H = \frac{\exp\left(\left\langle \boldsymbol{q}_i^H, \boldsymbol{k}_{j\to i}^H \right\rangle^{\kappa_H}_{\boldsymbol{p}_i^H} \right)}{\sum_{v_m \in \mathcal{N}(v_i)} \exp\left(\left\langle \boldsymbol{q}_i^H, \boldsymbol{k}_{m\to i}^H \right\rangle^{\kappa_H}_{\boldsymbol{p}_i^H} \right)}, \tag{26}$$

and the values are aggregated in the same tangent space using these weights:

$$\boldsymbol{z}_i^H = \sum_{v_j \in \mathcal{N}(v_i)} \beta_{ij}^H \boldsymbol{v}_{j\to i}^H, \qquad \boldsymbol{z}_i^H \in T_{\boldsymbol{p}_i^H} \mathcal{H}^d. \tag{27}$$

Similarly, the Spherical branch follows the same procedure.

To form the final node representation, we concatenate the structural positions and semantic representations from both geometric branches, defined as:

$$\boldsymbol{x}_i = \left[\boldsymbol{p}_i^H \,\|\, \boldsymbol{z}_i^H \,\|\, \boldsymbol{p}_i^S \,\|\, \boldsymbol{z}_i^S \right]. \tag{28}$$

3.5 Model Optimization

The learned geometric representations should match local graph distances, while semantic representations are aligned across geometries. To this end, we train R-MESD with two losses, namely a graph geometry consistency loss and a cross-geometry semantic contrastive loss.

Graph Geometry Consistency Loss. For a target node v_i, we construct a subgraph $G_{\mathcal{B}}$ by multi-hop neighbor sampling. Mathematically, let $d_{G_{\mathcal{B}}}(i,j)$ denote the shortest-path hop count between nodes v_i and v_j in this subgraph. The geodesic distance is defined as $d_{\mathrm{geo}}(i,j) = \sqrt{(d_{ij}^H)^2 + (d_{ij}^S)^2}$, where the branch-specific distances are given in Appendix H. The structural loss can be given as:

$$\mathcal{L}_{\mathrm{str}} = \frac{1}{|\Omega_{\mathcal{B}}|} \sum_{(i,j)\in\Omega_{\mathcal{B}}} \left| \frac{d_{G_{\mathcal{B}}}(i,j)}{d_{\mathrm{geo}}(i,j) + \varepsilon} - 1 \right|. \tag{29}$$

This objective aligns the learned geometric distances with local graph distances.

Cross-Geometry Semantic Contrastive Loss. To compare semantic representations across geometries, semantic vectors are parallel transported to the respective origin tangent spaces, yielding common coordinates. The Hyperbolic-to-Spherical loss is represented as:

$$\mathcal{J}(H,S) = -\frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \log \frac{\exp\left(\left\langle \mathrm{PT}^{\kappa_H}_{\boldsymbol{p}_i^H \to \boldsymbol{o}}(\boldsymbol{z}_i^H), \mathrm{PT}^{\kappa_S}_{\boldsymbol{p}_i^S \to \boldsymbol{o}}(\boldsymbol{z}_i^S) \right\rangle \right)}{\sum_{j\in\mathcal{B}} \exp\left(\left\langle \mathrm{PT}^{\kappa_H}_{\boldsymbol{p}_i^H \to \boldsymbol{o}}(\boldsymbol{z}_i^H), \mathrm{PT}^{\kappa_S}_{\boldsymbol{p}_j^S \to \boldsymbol{o}}(\boldsymbol{z}_j^S) \right\rangle \right)}. \tag{30}$$

Swapping the two branches gives $\mathcal{J}(S, H)$, and the symmetric semantic loss is:

$$\mathcal{L}_{\text{sem}} = \mathcal{J}(H, S) + \mathcal{J}(S, H). \tag{31}$$

The total training objective is:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{str}}\mathcal{L}_{\text{str}}, \tag{32}$$

where λ_{sem} and λ_{str} balance the two losses.

4 EXPERIMENTS

We conduct experiments to address the following research questions:

- **RQ1:** How does R-MESD perform compared with state-of-the-art baselines? (4.2)
- **RQ2:** How do the key components of R-MESD contribute to its overall performance? (4.3)
- **RQ3:** Can R-MESD address this mismatch and yield discriminative node representations? (4.4)

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate R-MESD on five benchmark datasets, namely Cora (Sen et al., 2008), Citeseer (Giles et al., 1998), PubMed (Sen et al., 2008), Arxiv-2023 (He et al., 2024), and OGBN-ArXiv (Hu et al., 2020). These datasets are structured as citation graphs, where nodes represent academic papers and edges denote citation relationships. Each dataset is randomly partitioned into training, validation, and test subsets with a 6:2:2 ratio. Dataset statistics are detailed in Appendix F.

Baselines. We compare against two groups of baselines. Traditional methods include MLP, GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), GraphSAGE (Hamilton et al., 2017), and GCNII (Chen et al., 2020). Other recent works designed for TAGs include G2P2 (Wen & Fang, 2023), SimTEG (Duan et al., 2023), CGT (Vakil & Amiri, 2024), GLEM (Zhao et al., 2023), TANS (Wang et al., 2025d), and LLM4NG (Yu et al., 2025). The 2026 methods include AD-GCN (Hevapathige et al., 2026), DeloopSGNN (Li et al., 2026), AdaSpec (Liu & Wang, 2026), RADE (Saber & Salehi-Abari, 2026), and SLGNN (Zhang et al., 2026). All methods are evaluated under the same data splits and reporting protocol.

4.2 PERFORMANCE COMPARISON (RQ1)

We compare R-MESD with traditional GNNs and recent methods designed for TAGs. As shown in Table 1, R-MESD achieves the highest accuracy on all five datasets, demonstrating its effectiveness. Specifically, on the compact Cora graph, R-MESD exceeds GAT by 1.98%, indicating that adaptive geometric experts offer benefits beyond Euclidean neighborhood attention. On the sparser Citeseer graph, R-MESD exceeds AD-GCN by 2.00%, highlighting the value of structural-semantic decoupling and cross-geometry semantic alignment when neighborhood information is limited. On PubMed, which provides richer node text, R-MESD separately encodes node semantics and refines them through tangent-space semantic attention, achieving a 2.46% improvement over DeloopSGNN. On ArXiv-2023, which covers a broader range of target classes, R-MESD outperforms GCN by 3.60%, suggesting that structural-semantic decoupling helps retain discriminative node semantics across diverse classes. On the substantially larger OGBN-ArXiv, R-MESD maintains a 2.44% lead over LLM4NG. Overall, these results indicate that R-MESD effectively mitigates the limitations of fixed neighborhood propagation under structural-semantic mismatch while accommodating diverse local structures beyond a single geometric space.

4.3 ABLATION STUDY (RQ2)

Table 2 compares R-MESD with its ablated variants across three datasets. Specifically, 1) removing the graph geometry consistency loss, $\mathcal{L}_{\text{str}}$, causes the largest decline on Cora (5.29%); 2) removing the cross-geometry semantic contrastive loss, $\mathcal{L}_{\text{sem}}$, has the strongest effect on PubMed (6.06%); 3) the w/o Cross-Geometry Structural Attention variant uses intra-branch attention and yields the largest drop on Cora (6.65%); 4) the w/o Cross-Geometry Semantic Alignment variant uses branch-wise processing, with its largest decline on PubMed (4.88%) and a decrease on ArXiv-2023 (3.66%); 5) Uniform Routing assigns equal expert weights and reduces performance most on Cora (5.29%); as well as 6) ALT Replacement uses a Log–Linear–Exp mapping instead of ALT and gives the largest drop on Cora (5.47%), underscoring the value of learned routing and geometric transformation.

Table 1: **Experimental results of node classification.** We report the mean Accuracy (%) with a standard deviation of 5 runs with different random seeds.

Methods	Cora	Citeseer	PubMed	Arxiv-2023	OGBN-ArXiv
GraphSAGE	85.56 ± 1.54	57.03 ± 1.26	82.35 ± 0.60	37.58 ± 4.52	32.22 ± 5.49
MLP	84.13 ± 0.31	69.99 ± 0.31	39.41 ± 0.90	70.20 ± 0.33	66.45 ± 0.33
GCN	84.97 ± 0.34	71.97 ± 0.33	50.11 ± 2.04	74.38 ± 0.33	70.76 ± 0.16
GCNII	88.51 ± 0.52	77.14 ± 1.21	89.54 ± 0.35	55.93 ± 2.52	58.36 ± 2.73
GAT	90.39 ± 0.46	72.74 ± 0.42	53.50 ± 1.15	70.46 ± 0.20	68.63 ± 0.11
SimTEG	78.41 ± 0.40	66.19 ± 0.29	74.17 ± 0.12	63.31 ± 0.14	63.10 ± 0.40
CGT	89.26 ± 0.45	68.02 ± 0.70	78.72 ± 0.55	55.82 ± 0.85	57.40 ± 0.77
G2P2	83.26 ± 2.39	73.07 ± 1.02	81.29 ± 0.11	52.33 ± 0.26	63.89 ± 0.01
GLEM	87.11 ± 1.14	74.53 ± 1.66	85.24 ± 2.70	60.43 ± 3.95	63.35 ± 0.21
SLGNN	87.16 ± 0.88	73.45 ± 1.72	88.33 ± 0.52	64.96 ± 1.01	61.11 ± 0.24
DeloopSGNN	88.38 ± 0.86	76.70 ± 2.03	90.13 ± 0.17	59.58 ± 0.46	61.01 ± 0.31
AdaSpec	87.97 ± 1.40	76.72 ± 1.50	86.99 ± 0.47	58.50 ± 0.25	67.20 ± 0.26
LLM4NG	87.55 ± 1.17	75.80 ± 1.34	88.61 ± 0.22	70.78 ± 0.35	73.67 ± 0.26
AD-GCN	86.90 ± 1.54	79.65 ± 1.18	88.99 ± 0.16	63.93 ± 0.89	65.87 ± 0.59
RADE	87.75 ± 0.89	75.50 ± 1.72	89.43 ± 0.41	64.68 ± 0.43	68.44 ± 0.31
TANS	88.80 ± 0.43	77.30 ± 0.91	88.99 ± 0.30	67.78 ± 0.38	69.57 ± 0.34
R-MESD	**92.37 ± 0.31**	**81.65 ± 1.17**	**92.59 ± 0.62**	**77.98 ± 1.62**	**76.11 ± 1.45**

Table 2: **Ablation study.** We report mean Accuracy (%) across three datasets.

Variants	Cora	PubMed	ArXiv-2023
w/o $\mathcal{L}_{\text{str}}$	87.08	90.66	74.56
w/o $\mathcal{L}_{\text{sem}}$	86.53	86.53	73.60
w/o Cross-Geometry Structural Attention	85.72	90.89	75.76
w/o Cross-Geometry Semantic Alignment	88.56	87.71	74.32
Uniform Routing	87.08	89.88	74.85
ALT Replacement	86.90	89.04	74.87
R-MESD	**92.37**	**92.59**	**77.98**

4.4 Representation Visualization (RQ3)

Figure 3 presents the t-SNE (van der Maaten & Hinton, 2008) visualizations on Cora for (a) Raw Features, (b) W/o $\mathcal{L}_{\text{sem}}$, and (c) R-MESD. Compared with the two baselines, R-MESD yields clearer class separation, indicating more discriminative node representations.

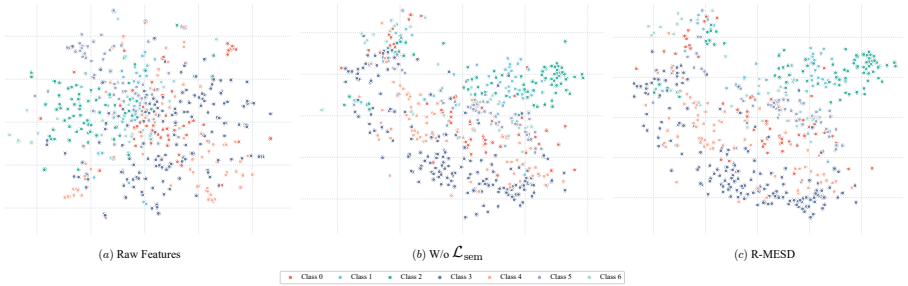


Figure 3: t-SNE visualization on Cora for (a) Raw Features, (b) W/o $\mathcal{L}_{\text{sem}}$, and (c) R-MESD.

5 Conclusion

In this paper, we present R-MESD, a Riemannian Multi-Expert Structural-Semantic Decoupling model for addressing structural-semantic mismatch and heterogeneous local topologies in TAGs. R-MESD decouples structural and semantic representations by modeling local topology with adaptively weighted Hyperbolic and Spherical experts and encoding node semantics as tangent vectors anchored at learned structural positions. Cross-geometry structural and semantic interactions further integrate complementary information across geometric views and local neighborhoods. Extensive experiments on five benchmark datasets demonstrate the effectiveness of our proposed R-MESD over existing state-of-the-art methods.

AI USE STATEMENT

In this work, we used generative AI tools to support language refinement, including grammar correction, sentence restructuring, terminology consistency, and improvements to the clarity, coherence, and presentation of the manuscript. We also used generative AI tools for limited programming assistance during implementation, code inspection, debugging, and documentation. Generative AI tools were not used to develop the core methodology, formulate the theoretical claims, generate or fabricate experimental results, modify experimental data, or draw scientific conclusions without author verification. All methodological decisions, mathematical derivations, experimental designs, data analyses, and interpretations were made and checked by the authors.

REPRODUCIBILITY STATEMENT

We support reproducibility by providing a detailed description of the R-MESD method, training procedure, benchmark datasets, baseline configurations, and evaluation protocol throughout the paper. Appendix F reports the dataset statistics, model hyperparameters, sampling settings, optimization details, and other implementation choices needed to reproduce the experiments. Upon formal publication, we will release the complete source code, preprocessing scripts, configuration files, generated auxiliary inputs, dataset split files, and detailed instructions for reproducing all reported experiments.

REFERENCES

Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 486–496, 2020.

Jianyuan Bo, Hao Wu, and Yuan Fang. Quantizing text-attributed graphs for semantic-structural integration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 107–118, 2025.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32*, pp. 4869–4880, 2019.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1725–1735, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.

Zihan Fang, Zhiling Cai, Yuxuan Zheng, Shide Du, Yanchao Tan, and Shiping Wang. Hituner: Hierarchical semantic fusion model fine-tuning on text-attributed graphs. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pp. 5110–5117, 2025.

C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the ACM International Conference on Digital Libraries*, pp. 89–98, 1998.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Asela Hevapathige, Asiri Wijesinghe, and Ahad N. Zehmakan. Beyond fixed depth: Adaptive graph neural networks for node classification under varying homophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21717–21725, 2026.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Proceedings of Advances in Neural Information Processing Systems*, 2020.

Ilia Igashov, Nikita Pavlichenko, and Sergei Grudinin. Spherical convolutions on molecular graphs for protein model quality assessment. *Mach. Learn. Sci. Technol.*, 2(4):045005, 2021.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Duanyu Li, Huijun Wu, Min Xie, Kai Lu, Wenzhe Zhang, Zhenwei Wu, Yong Dong, and Ruibo Wang. Deloopsgnn: Revisiting spectral gnns through the lens of spatial aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 22878–22886, 2026.

Shujie Li, Yuxia Wu, Chuan Shi, and Yuan Fang. Hetgb: A comprehensive benchmark for heterophilic text-attributed graphs. *arXiv preprint arXiv:2503.04822*, 2025.

Fangbing Liu and Qing Wang. Adaspec: Adaptive spectrum for enhanced node distinguishability. In *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *Proceedings of 8th International Conference on Learning Representations*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3980–3990, 2019.

Danial Saber and Amirali Salehi-Abari. Rade: Random add-drop edge as a regularizer. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Guangxin Su, Hanchen Wang, Jianwei Wang, Wenjie Zhang, Ying Zhang, and Jian Pei. Large language models meet text-attributed graphs: A survey of integration frameworks and applications. *arXiv preprint arXiv:2510.21131*, 2025.

Li Sun and Philip S. Yu. A riemannian perspective on graph foundation models: curvature as a guiding principle. *Frontiers Comput. Sci.*, 20(12):2012370, 2026.

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

Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip S. Yu. Riemanngfm: Learning a graph foundation model from riemannian geometry. In *Proceedings of the ACM on Web Conference 2025*, pp. 1154–1165, 2025.

Nidhi Vakil and Hadi Amiri. Controlled transformation of text-attributed graphs. In *Proceedings of Findings of the Association for Computational Linguistics: EMNLP*, pp. 15735–15748, 2024.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2605):2579–2605, 2008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. Text meets topology: Rethinking out-of-distribution detection in text-rich networks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5494–5523, 2025a.

Haoyu Peter Wang, Shikun Liu, Rongzhe Wei, and Pan Li. Generalization principles for inference over text-attributed graphs with large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

Yaoke Wang, Yun Zhu, Wenqiao Zhang, Yueting Zhuang, Liyunfei, and Siliang Tang. Bridging local details and global context in text-attributed graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14830–14841, 2024.

Yuxiang Wang, Xiao Yan, Shiyu Jin, Quanqing Xu, Chuang Hu, Yuanyuan Zhu, Bo Du, Jia Wu, and Jiawei Jiang. Exploiting text semantics for few and zero shot node classification on text-attributed graph. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pp. 3462–3470, 2025c.

Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. Can llms convert graphs to text-attributed graphs? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1412–1432, 2025d.

Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 506–516, 2023.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 17238–17264, 2023.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 2111–2117, 2015.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 28798–28810, 2021.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13087–13095, 2025.

Kaicheng Zhang, David N. Reynolds, Piero Deidda, and Francesco Tudisco. Stuart-landau oscillatory graph neural network. In *Proceedings of the ACM Web Conference 2026*, pp. 1469–1480, 2026.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Chuang Zhou, Jiahe Du, Huachi Zhou, Hao Chen, Feiran Huang, and Xiao Huang. Text-attributed graph learning with coupled augmentations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10865–10876, 2025a.

Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. Taming language models for text-attributed graph learning with decoupled aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 3463–3474, 2025b.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Proceedings of Advances in Neural Information Processing Systems*, 2020.

A Additional Experiments

A.1 Geometric Configuration Ablation Study

We compare R-MESD with five alternative geometric branch configurations in Table 3. With the H / S configuration, R-MESD achieves the highest Accuracy on all three datasets. On Citeseer, it exceeds the strongest alternative by 1.72%. The gains reach 2.97% on PubMed and 3.16% on Arxiv-2023. Replacing the Hyperbolic branch with a Euclidean one reduces accuracy by 6.06% on PubMed. Using H / H or S / S also leads to drops of 5.37% and 5.27% on PubMed, respectively. The E / E configuration causes the largest drop on Citeseer (8.45%). These results suggest that Hyperbolic and Spherical geometries complement each other in node representation learning.

Table 3: **Geometric ablation on Citeseer, PubMed, and Arxiv-2023.** H, S, and E denote Hyperbolic, Spherical, and Euclidean geometries, respectively. We report the mean Accuracy (%) with a standard deviation of 5 runs with different random seeds.

Geometric Branches	Citeseer	PubMed	Arxiv-2023
H / S	**81.65 ± 1.17**	**92.59 ± 0.62**	**77.98 ± 1.62**
H / E	79.08 ± 0.07	89.62 ± 1.21	74.77 ± 1.98
S / E	79.93 ± 1.66	86.53 ± 1.08	74.79 ± 2.04
H / H	78.22 ± 2.31	87.22 ± 2.72	74.74 ± 1.62
S / S	78.37 ± 1.05	87.32 ± 0.87	74.64 ± 0.55
E / E	73.20 ± 0.35	87.04 ± 2.56	74.82 ± 0.89

A.2 Heterophily-Grouped Performance Comparison

To examine how R-MESD responds to increasing structural-semantic mismatch, we divide the test nodes into four heterophily intervals. R-MESD is compared with the second-best method in Table 1. As shown in Table 4, R-MESD achieves clear gains over the second-best method in the two highest-heterophily intervals on all five datasets. In the 0.75–1.0 interval, the improvements reach 11.30% on Cora, 7.77% on Citeseer, 8.68% on PubMed, 3.62% on Arxiv-2023, and 2.51% on OGBN-ArXiv. Moreover, R-MESD outperforms the second-best method across all four intervals on Arxiv-2023 and OGBN-ArXiv. Although its performance is slightly lower in the lower-heterophily intervals on Cora, Citeseer, and PubMed, the gains emerge and widen in the higher-heterophily intervals. These results show that R-MESD is particularly effective when fixed neighborhood propagation is more likely to introduce structural-semantic interference.

Table 4: **Heterophily-grouped node classification results.** R-MESD is compared with the second-best method for each dataset. We report mean Accuracy (%) across five datasets. In the R-MESD rows, a red upward arrow denotes an improvement over the corresponding method, while a green downward arrow denotes a decrease.

		Heterophily Intervals			
Datasets	**Methods**	**0–0.25**	**0.25–0.5**	**0.5–0.75**	**0.75–1.0**
Cora	GAT	93.57	89.68	80.46	75.64
	R-MESD	93.43 ↓ 0.14	92.36 ↑ 2.68	88.75 ↑ 8.29	86.94 ↑ 11.30
Citeseer	AD-GCN	88.97	82.37	75.74	68.88
	R-MESD	87.09 ↓ 1.88	81.84 ↓ 0.53	78.92 ↑ 3.18	76.65 ↑ 7.77
PubMed	DeloopSGNN	97.23	93.43	88.61	79.73
	R-MESD	96.91 ↓ 0.32	92.86 ↓ 0.57	90.56 ↑ 1.95	88.41 ↑ 8.68
Arxiv-2023	GCN	84.79	79.15	72.26	65.44
	R-MESD	88.39 ↑ 3.60	82.73 ↑ 3.58	75.85 ↑ 3.59	69.06 ↑ 3.62
OGBN-ArXiv	LLM4NG	84.29	76.15	70.39	66.86
	R-MESD	86.58 ↑ 2.29	79.06 ↑ 2.91	72.65 ↑ 2.26	69.37 ↑ 2.51

B Related Work

We concisely discuss closely relevant researches, including representation learning on TAGs, Hyperbolic and Spherical graph neural networks, and mixed-curvature graph representation learning.

B.1 Representation Learning on TAGs

Early approaches to representation learning on TAGs typically treat textual information as static node features and propagate these features along the given graph topology (Yang et al., 2015; 2021; Yan et al., 2023). Representative models, including GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), and GraphSAGE (Hamilton et al., 2017), learn node representations through neighborhood aggregation. Although effective at incorporating structural context, these methods do not explicitly model the semantic dependencies within raw node text. To bridge this gap, subsequent studies combine pretrained language models (PLMs) with GNNs, often using encoders such as BERT (Devlin et al., 2019). Specifically, a common two-stage strategy first adapts a language model to generate text embeddings and then uses these embeddings as GNN inputs. For instance, SimTEG employs parameter-efficient fine-tuning to obtain text representations for downstream graph learning (Duan et al., 2023). Furthermore, joint learning strategies enable closer interaction between the language and graph modules. GLEM adopts a variational expectation-maximization framework to alternate between optimizing the two modules, allowing them to exchange predictive information (Zhao et al., 2023). In addition, LLM4NG uses large language models to generate labeled nodes and augment the graph for few-shot learning (Yu et al., 2025). Despite these advances, stronger text encoders alone do not ensure that semantic relations agree with local connectivity. When structural-semantic mismatch occurs, neighborhood propagation can introduce irrelevant semantic information. Moreover, a single Euclidean representation space may not adequately capture heterogeneous local topologies. These limitations motivate structural-semantic decoupling together with adaptive geometric representations.

Building on these directions, recent TAG studies explore hierarchical semantic fusion for fine-tuning (Fang et al., 2025), exploit text semantics for few-shot node classification (Wang et al., 2025c), and analyze generalization principles for LLM-based inference (Wang et al., 2025b). Other work quantizes text-attributed graphs to integrate semantic and structural information (Bo et al., 2025), while recent analysis of text-rich networks examines the interaction between text and topology (Wang et al., 2025a). Together, these studies broaden the use of textual information in TAGs, yet they do not directly resolve the mismatch between semantic relations and local graph connectivity.

B.2 Hyperbolic and Spherical Graph Neural Networks

Non-Euclidean graph neural networks exploit geometric properties that better match particular graph structures. In particular, the exponential volume growth of Hyperbolic space makes it well suited to tree-like and hierarchical structures. Hyperbolic GCN extends feature transformation and neighborhood aggregation to Hyperbolic space, thereby exploiting its hierarchical inductive bias for graph learning (Chami et al., 2019). Meanwhile, Spherical geometry provides a complementary setting for representing cyclic patterns, as illustrated by Spherical graph convolutions (Igashov et al., 2021). To support graph convolution in these different spaces, a unified constant-curvature framework based on the κ-stereographic model defines feature transformations and aggregation for Hyperbolic, Spherical, and Euclidean geometries, with extensions to their product spaces (Bachmann et al., 2020). These developments highlight the importance of matching representation geometry to graph structure (Pei et al., 2020). Nevertheless, a single constant-curvature space, even with learnable curvature, can be insufficient for representing heterogeneous local topologies. This motivates multiple geometric experts whose contributions adapt to local structural patterns.

B.3 Mixed-Curvature Graph Representation Learning

Mixed-curvature methods combine complementary geometries to model diverse structural patterns. In this way, graph representations can extend beyond a single space. Specifically, graph convolution has been extended to products of constant-curvature spaces, allowing component curvatures to be learned from data. Building on this direction, SelfMGNN combines multiple Riemannian components through hierarchical attention and contrastive learning (Sun et al., 2022). More recently, RiemannGFM constructs a product bundle that couples manifold coordinates with tangent-space

node encodings (Sun et al., 2025). It encodes a structural vocabulary of trees and cycles using cross-geometry attention and tangent-space aggregation, supporting graph representation learning across domains. Related work further studies graph foundation models from a Riemannian perspective, treating curvature as a guiding principle (Sun & Yu, 2026). Taken together, these studies demonstrate the value of combining geometries and coordinating their representations. However, the reviewed methods primarily address geometric composition and cross-space fusion, while the joint modeling of structural-semantic mismatch and heterogeneous local topologies in TAGs remains less explored. Consequently, jointly aligning manifold-based structural representations with text-derived semantics in TAGs remains an underexplored problem.

C Theoretical Proofs

C.1 Proof of Theorem 1

Proof. Fix $\kappa \neq 0$, $\varepsilon > 0$, and $\rho \in (0, 1)$ as in Theorem 1. The proof establishes that the initialized structural positions lie in $\mathcal{M}_\kappa^d$ and that their coordinate differences satisfy the stated Lipschitz bound. To separate normalization from curvature-dependent scaling, let $r_\kappa = \rho/\sqrt{|\kappa|}$ and define the normalization map ϕ and initialization map F_κ as follows:

$$\phi(\boldsymbol{s}) = \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|_2 + \varepsilon}, \qquad F_\kappa(\boldsymbol{s}) = r_\kappa \phi(\boldsymbol{s}).$$

Since the denominator is strictly positive, both maps are well defined for every spectral representation $\boldsymbol{s} \in \mathbb{R}^d$, including $\boldsymbol{s} = \mathbf{0}$.

First, the initialization map F_κ gives the following norm bound for each initialized structural position:

$$\|F_\kappa(\boldsymbol{s})\|_2 = r_\kappa \frac{\|\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2 + \varepsilon} < r_\kappa = \frac{\rho}{\sqrt{|\kappa|}}.$$

For the Hyperbolic case $\kappa < 0$, the κ-stereographic coordinate domain requires $1 + \kappa\|\boldsymbol{x}\|_2^2 > 0$. Substituting this bound into the domain condition yields:

$$1 + \kappa\|F_\kappa(\boldsymbol{s})\|_2^2 = 1 - |\kappa|\|F_\kappa(\boldsymbol{s})\|_2^2 > 1 - \rho^2 > 0.$$

Thus, the initialized positions remain strictly inside the Hyperbolic coordinate domain. For the Spherical case $\kappa > 0$, this coordinate domain is all of $\mathbb{R}^d$, since $1 + \kappa\|\boldsymbol{x}\|_2^2 \geq 1$. Consequently, $F_\kappa(\boldsymbol{s}) \in \mathcal{M}_\kappa^d$ holds for both geometric branches.

To establish the second property, we bound the Jacobian of the normalization map. For $\boldsymbol{s} \neq \mathbf{0}$, let $a = \|\boldsymbol{s}\|_2$ and $\boldsymbol{e} = \boldsymbol{s}/a$. Differentiating ϕ and grouping the terms by direction gives the following decomposition:

$$\begin{aligned} \boldsymbol{J}_\phi(\boldsymbol{s}) &= \frac{1}{a+\varepsilon}\boldsymbol{I} - \frac{\boldsymbol{s}\boldsymbol{s}^\top}{a(a+\varepsilon)^2} \\ &= \frac{\varepsilon}{(a+\varepsilon)^2}\boldsymbol{e}\boldsymbol{e}^\top + \frac{1}{a+\varepsilon}\left(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^\top\right), \end{aligned}$$

where $\boldsymbol{e}\boldsymbol{e}^\top$ and $\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^\top$ are orthogonal projections onto the direction of $\boldsymbol{s}$ and its orthogonal complement, respectively. The two coefficients therefore describe how normalization scales perturbations in these directions. In particular, the coefficients satisfy the following inequalities:

$$0 < \frac{\varepsilon}{(a+\varepsilon)^2} \leq \frac{1}{a+\varepsilon} \leq \frac{1}{\varepsilon}.$$

It follows that $\|\boldsymbol{J}_\phi(\boldsymbol{s})\|_2 \leq 1/\varepsilon$ for every nonzero $\boldsymbol{s}$, where the matrix norm is the Euclidean operator norm.

The origin requires a separate check because the preceding expression uses $a > 0$. For any nonzero perturbation $\boldsymbol{h}$, subtracting the linear term $\boldsymbol{h}/\varepsilon$ gives the following normalized remainder:

$$\frac{\|\phi(\boldsymbol{h}) - \phi(\mathbf{0}) - \boldsymbol{h}/\varepsilon\|_2}{\|\boldsymbol{h}\|_2} = \frac{\|\boldsymbol{h}\|_2}{\varepsilon(\|\boldsymbol{h}\|_2 + \varepsilon)} \longrightarrow 0 \qquad \text{as } \boldsymbol{h} \to \mathbf{0}.$$

Hence, ϕ is differentiable at the origin with $\boldsymbol{J}_\phi(\mathbf{0}) = \boldsymbol{I}/\varepsilon$. Moreover, the rank-one term in the first Jacobian expression has norm $a/(a+\varepsilon)^2$, which tends to zero as $a \to 0$. The Jacobian therefore converges to $\boldsymbol{I}/\varepsilon$ at the origin, so ϕ is continuously differentiable on $\mathbb{R}^d$. Combining the two cases, we obtain the uniform bound:

$$\|\boldsymbol{J}_\phi(\boldsymbol{s})\|_2 \le \frac{1}{\varepsilon}, \qquad \boldsymbol{s} \in \mathbb{R}^d.$$

Finally, consider arbitrary $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^d$ and the line segment $\boldsymbol{\gamma}(\tau) = \boldsymbol{t} + \tau(\boldsymbol{s} - \boldsymbol{t})$, where $\tau \in [0,1]$. The continuous Jacobian allows the fundamental theorem of calculus to be applied along this segment, even when it passes through the origin:

$$\phi(\boldsymbol{s}) - \phi(\boldsymbol{t}) = \int_0^1 \boldsymbol{J}_\phi(\boldsymbol{\gamma}(\tau))(\boldsymbol{s} - \boldsymbol{t})\, d\tau.$$

Taking norms and applying the uniform Jacobian bound gives:

$$\begin{aligned}\|F_\kappa(\boldsymbol{s}) - F_\kappa(\boldsymbol{t})\|_2 &= r_\kappa \|\phi(\boldsymbol{s}) - \phi(\boldsymbol{t})\|_2 \\ &\le r_\kappa \int_0^1 \|\boldsymbol{J}_\phi(\boldsymbol{\gamma}(\tau))\|_2 \|\boldsymbol{s} - \boldsymbol{t}\|_2\, d\tau \\ &\le \frac{r_\kappa}{\varepsilon} \|\boldsymbol{s} - \boldsymbol{t}\|_2.\end{aligned}$$

Thus, for each fixed expert curvature, the initialization map is globally r_κ/ε-Lipschitz in the Euclidean coordinate norm. Together with the domain argument, this proves both properties in Theorem 1.

C.2 Proof of Theorem 2

Proof. Let $\boldsymbol{x} = [x_t; \boldsymbol{x}_s] \in \mathcal{L}_\kappa^{d_\kappa}$ and $\boldsymbol{W}_\kappa \in \mathbb{R}^{d \times d_\kappa}$, with $\kappa, \bar{\kappa} \neq 0$ as in Theorem 2. The proof first verifies that the temporal correction factor compensates for the spatial transformation, so that the transformed representation satisfies the target spacetime constraint. The forward stereographic projection then returns its manifold coordinates under the theorem's projection condition.

At the outset, the source representation satisfies the spacetime constraint in Eq. 10. Using the spacetime form in Eq. 11, this constraint relates the temporal and spatial coordinates as follows:

$$\begin{aligned}\operatorname{sgn}(\kappa) x_t^2 + \|\boldsymbol{x}_s\|_2^2 &= \frac{1}{\kappa}, \\ 1 - \kappa \|\boldsymbol{x}_s\|_2^2 &= |\kappa| x_t^2.\end{aligned}$$

Since the temporal correction factor is assumed to be well defined, its denominator $1 - \kappa\|\boldsymbol{x}_s\|_2^2$ is nonzero. The second identity therefore shows that this denominator is strictly positive and expresses it in terms of the source temporal coordinate.

Subsequently, the learnable spatial transformation maps $\boldsymbol{x}_s$ to $\mathbb{R}^d$. Define the squared spatial norm $\ell_\kappa(\boldsymbol{x}_s)$ after this transformation as follows:

$$\ell_\kappa(\boldsymbol{x}_s) = \|\boldsymbol{W}_\kappa \boldsymbol{x}_s\|_2^2.$$

To satisfy the target constraint after this transformation, ALT rescales the temporal coordinate using the correction factor in Eq. 13:

$$\gamma_\kappa(\boldsymbol{x}_s) = \sqrt{\frac{|\kappa|}{|\bar{\kappa}|} \frac{1 - \bar{\kappa}\ell_\kappa(\boldsymbol{x}_s)}{1 - \kappa\|\boldsymbol{x}_s\|_2^2}}.$$

Under the stated assumption, the square root is well defined. In particular, $1 - \bar{\kappa}\ell_\kappa(\boldsymbol{x}_s) \ge 0$; this holds automatically for a Hyperbolic target, while a Spherical target requires $\ell_\kappa(\boldsymbol{x}_s) \le 1/\bar{\kappa}$. Combining the spatial transformation with this temporal correction gives the ALT output:

$$\boldsymbol{y} = [y_t; \boldsymbol{y}_s] = \mathrm{ALT}_{\kappa \to \bar{\kappa}}(\boldsymbol{W}_\kappa, \boldsymbol{x}) = \begin{bmatrix} \gamma_\kappa(\boldsymbol{x}_s) x_t \\ \boldsymbol{W}_\kappa \boldsymbol{x}_s \end{bmatrix}.$$

Substituting the source identity for x_t^2 into the temporal update cancels the factors involving the source curvature. Consequently, the temporal and spatial components satisfy:

$$\begin{aligned} y_t^2 &= \gamma_\kappa(\boldsymbol{x}_s)^2 x_t^2 \\ &= \frac{|\kappa|}{|\bar{\kappa}|} \frac{1-\bar{\kappa}\ell_\kappa(\boldsymbol{x}_s)}{1-\kappa\|\boldsymbol{x}_s\|_2^2} \frac{1-\kappa\|\boldsymbol{x}_s\|_2^2}{|\kappa|} \\ &= \frac{1-\bar{\kappa}\ell_\kappa(\boldsymbol{x}_s)}{|\bar{\kappa}|}, \\ \|\boldsymbol{y}_s\|_2^2 &= \ell_\kappa(\boldsymbol{x}_s). \end{aligned}$$

These identities show how the temporal correction accounts for the transformed spatial norm and the target curvature. Substituting them into the target spacetime form and using $\operatorname{sgn}(\bar{\kappa})/|\bar{\kappa}| = 1/\bar{\kappa}$ reduces the resulting quadratic form to $1/\bar{\kappa}$, thereby establishing the required constraint:

$$\begin{aligned} \langle \boldsymbol{y}, \boldsymbol{y} \rangle_{\bar{\kappa}} &= \operatorname{sgn}(\bar{\kappa}) y_t^2 + \|\boldsymbol{y}_s\|_2^2 \\ &= \frac{1-\bar{\kappa}\ell_\kappa(\boldsymbol{x}_s)}{\bar{\kappa}} + \ell_\kappa(\boldsymbol{x}_s) \\ &= \frac{1}{\bar{\kappa}}. \end{aligned}$$

Moreover, $\boldsymbol{y}_s \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$, so the output has the required dimension. Thus, $\boldsymbol{y} \in \mathcal{L}_{\bar{\kappa}}^d$ for either sign of the target curvature, proving the first conclusion.

Finally, under the assumed validity of the forward stereographic projection on the target coordinate domain, Eq. 14 gives the transformed structural representation:

$$\Pi_{\bar{\kappa}}(\boldsymbol{y}) = \frac{\boldsymbol{y}_s}{1+\sqrt{|\bar{\kappa}|}\, y_t} \in \mathcal{M}_{\bar{\kappa}}^d.$$

This establishes the second conclusion under the stated projection condition. Therefore, ALT aligns curvature and spatial dimensionality while satisfying the target manifold constraint.

D Basic Riemannian Operations

Here we collect coordinate expressions for the quantities and operations used in the κ-stereographic model. That is, formulas with $\sqrt{|\kappa|}$ in a denominator are stated for $\kappa \neq 0$, their Euclidean counterparts follow by taking the limit as $\kappa \to 0$. Expressions with a vanishing norm, such as $\boldsymbol{v} = \mathbf{0}$ or $\boldsymbol{x} = \boldsymbol{y}$, are interpreted by continuity.

For $\boldsymbol{x} \in \mathcal{M}_\kappa^d$, the conformal factor and Riemannian metric are:

$$\lambda_{\boldsymbol{x}}^\kappa = \frac{2}{1+\kappa\|\boldsymbol{x}\|_2^2}, \qquad \boldsymbol{g}_{\boldsymbol{x}}^\kappa = \left(\lambda_{\boldsymbol{x}}^\kappa\right)^2 \boldsymbol{I}. \tag{33}$$

For tangent vectors $\boldsymbol{a}, \boldsymbol{b} \in T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$, the Riemannian inner product induced by this metric is:

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\boldsymbol{x}}^\kappa = \left(\lambda_{\boldsymbol{x}}^\kappa\right)^2 \boldsymbol{a}^\top \boldsymbol{b}. \tag{34}$$

To express the maps below in stereographic coordinates, we define Möbius addition for $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}_\kappa^d$:

$$\boldsymbol{x} \oplus_\kappa \boldsymbol{y} = \frac{\left(1-2\kappa\boldsymbol{x}^\top\boldsymbol{y} - \kappa\|\boldsymbol{y}\|_2^2\right)\boldsymbol{x} + \left(1+\kappa\|\boldsymbol{x}\|_2^2\right)\boldsymbol{y}}{1-2\kappa\boldsymbol{x}^\top\boldsymbol{y} + \kappa^2\|\boldsymbol{x}\|_2^2\|\boldsymbol{y}\|_2^2}. \tag{35}$$

Let $\tan_\kappa(t)$ be $\tan(t)$, t, or $\tanh(t)$ for $\kappa > 0$, $\kappa = 0$, or $\kappa < 0$, respectively, and let $\tan_\kappa^{-1}$ denote its inverse. For $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$, the exponential map takes the following form:

$$\operatorname{Exp}_{\boldsymbol{x}}^\kappa(\boldsymbol{v}) = \boldsymbol{x} \oplus_\kappa \left(\tan_\kappa\left(\sqrt{|\kappa|} \frac{\lambda_{\boldsymbol{x}}^\kappa \|\boldsymbol{v}\|_2}{2} \right) \frac{\boldsymbol{v}}{\sqrt{|\kappa|}\|\boldsymbol{v}\|_2} \right). \tag{36}$$

Its inverse, the logarithmic map, is given by:

$$\operatorname{Log}_{\boldsymbol{x}}^\kappa(\boldsymbol{y}) = \frac{2}{\sqrt{|\kappa|}\lambda_{\boldsymbol{x}}^\kappa} \tan_\kappa^{-1}\left(\sqrt{|\kappa|}\, \| -\boldsymbol{x} \oplus_\kappa \boldsymbol{y}\|_2 \right) \frac{-\boldsymbol{x} \oplus_\kappa \boldsymbol{y}}{\| -\boldsymbol{x} \oplus_\kappa \boldsymbol{y}\|_2}. \tag{37}$$

In addition, parallel transport maps a tangent vector from $T_{\boldsymbol{x}}\mathcal{M}_\kappa^d$ to $T_{\boldsymbol{y}}\mathcal{M}_\kappa^d$ according to:

$$\mathrm{PT}_{\boldsymbol{x}\to\boldsymbol{y}}^\kappa(\boldsymbol{v}) = \frac{\lambda_{\boldsymbol{x}}^\kappa}{\lambda_{\boldsymbol{y}}^\kappa}\left(-\left(\boldsymbol{y}\oplus_\kappa(-\boldsymbol{x})\right)\oplus_\kappa\left(\boldsymbol{y}\oplus_\kappa\left((-\boldsymbol{x})\oplus_\kappa\boldsymbol{v}\right)\right)\right). \tag{38}$$

Using the exponential and logarithmic maps at the origin, scalar multiplication and matrix–vector multiplication are defined by:

$$r\otimes_\kappa\boldsymbol{x} = \mathrm{Exp}_{\boldsymbol{o}}^\kappa\left(r\,\mathrm{Log}_{\boldsymbol{o}}^\kappa(\boldsymbol{x})\right), \qquad \boldsymbol{W}\otimes_\kappa\boldsymbol{x} = \mathrm{Exp}_{\boldsymbol{o}}^\kappa\left(\boldsymbol{W}\,\mathrm{Log}_{\boldsymbol{o}}^\kappa(\boldsymbol{x})\right). \tag{39}$$

Finally, let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\in\mathcal{M}_\kappa^d$ and let the weights satisfy $\alpha_i\geq 0$ and $\sum_{i=1}^n\alpha_i>0$. When $\sum_{j=1}^n\alpha_j(\lambda_{\boldsymbol{x}_j}^\kappa-1)\neq 0$, the geometric centroid is:

$$m_\kappa(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n;\boldsymbol{\alpha}) = \frac{1}{2}\otimes_\kappa\left(\sum_{i=1}^n\frac{\alpha_i\lambda_{\boldsymbol{x}_i}^\kappa}{\sum_{j=1}^n\alpha_j\left(\lambda_{\boldsymbol{x}_j}^\kappa-1\right)}\boldsymbol{x}_i\right). \tag{40}$$

As $\kappa\to 0$, this centroid reduces to the weighted Euclidean average $\sum_{i=1}^n\alpha_i\boldsymbol{x}_i/\sum_{i=1}^n\alpha_i$.

E Local Structural Description

E.1 Local Evidence Construction

For each target node v_i, we combine structural statistics and textual context to construct routing evidence. Formally, let $\mathcal{N}(v_i)$ and $\mathcal{N}^{(2)}(v_i)$ denote its one-hop and strict two-hop neighborhoods, respectively, where:

$$\mathcal{N}^{(2)}(v_i) = \{u\in\mathcal{V}:\mathrm{dist}_{\mathcal{G}}(v_i,u)=2\}. \tag{41}$$

Thus, $\mathcal{N}^{(2)}(v_i)$ excludes v_i and its one-hop neighbors. We define the node degree as:

$$d_i = |\mathcal{N}(v_i)|. \tag{42}$$

The following scores use graph topology alone, without node labels. We first define the edges among the one-hop neighbors:

$$E_i^{\mathrm{nbr}} = \{(u,v)\in\mathcal{E}:u,v\in\mathcal{N}(v_i)\}. \tag{43}$$

We use neighborhood closure as a proxy for local cyclic tendency:

$$\rho_i^{\mathrm{cycle}} = \frac{2|E_i^{\mathrm{nbr}}|}{d_i(d_i-1)+\varepsilon}. \tag{44}$$

Then, we measure outward expansion by comparing the number of strict two-hop nodes with the degree:

$$b_i = \frac{|\mathcal{N}^{(2)}(v_i)|}{d_i+\varepsilon}. \tag{45}$$

Each two-hop node contributes once, even if several paths reach it. To limit the influence of large expansion ratios, we compute the 5th and 95th percentiles $q_{0.05}$ and $q_{0.95}$ of $\{b_i\}$ over all nodes. The clipping function can then be defined as:

$$\mathrm{clip}(x,a,b) = \min\{\max\{x,a\},b\}. \tag{46}$$

This gives the normalized expansion rate:

$$\widetilde{b}_i = \frac{\mathrm{clip}(b_i,q_{0.05},q_{0.95})-q_{0.05}}{q_{0.95}-q_{0.05}+\varepsilon}. \tag{47}$$

We combine normalized expansion with neighborhood closure to obtain the local branching score:

$$\rho_i^{\mathrm{tree}} = \widetilde{b}_i\left(1-\rho_i^{\mathrm{cycle}}\right). \tag{48}$$

A high score indicates strong outward expansion and low neighborhood closure.

After that, we capture degree differences between a node and its neighbors. Let the maximum degree in the graph be:

$$d_{\max} = \max_{v\in\mathcal{V}} d_v, \tag{49}$$

Using this graph-wide scaling value, we define a proxy for local hierarchical tendency:

$$\rho_i^{\text{hier}} = \frac{1}{d_i + \varepsilon} \sum_{v_j \in \mathcal{N}(v_i)} \frac{|\log(d_i + 1) - \log(d_j + 1)|}{\log(d_{\max} + 1) + \varepsilon}. \tag{50}$$

Larger values indicate greater degree disparity between the target node and its neighbors.

Finally, we add textual context by collecting texts from up to n_{text} neighbors of v_i:

$$n_i = \min(d_i, n_{\text{text}}), \qquad \mathcal{S}_i \subseteq \mathcal{N}(v_i), \quad |\mathcal{S}_i| = n_i, \quad \mathcal{T}_{\mathcal{N}(v_i)} = \{t_j : v_j \in \mathcal{S}_i\}. \tag{51}$$

Here, $\mathcal{T}_{\mathcal{N}(v_i)}$ contains the texts of the sampled neighbors in $\mathcal{S}_i$. Together, the structural scores and textual context form the evidence tuple:

$$\mathcal{Q}_i = \left(d_i, t_i, \mathcal{T}_{\mathcal{N}(v_i)}, \rho_i^{\text{tree}}, \rho_i^{\text{cycle}}, \rho_i^{\text{hier}}\right). \tag{52}$$

E.2 Prompt Template and Design Principles

We describe the language-model prompt used to generate local structural descriptions, where the system prompt, user prompt, and an input–output example are shown separately below.

System Prompt: You are responsible for generating a concise, objective, and verifiable description of a target node's local substructure from its local structural evidence and textual context.
Instruction:

1. Use the target node degree to describe local connectivity, without interpreting it as a property of the whole graph.
2. Use ρ_i^{tree}, ρ_i^{cycle}, and ρ_i^{hier} to characterize local structural tendencies; these quantities are structural proxies rather than strict structural judgments.
3. Use target-node and neighbor text only to supplement local semantic context, and never to infer topological properties.
4. When the structural evidence is conflicting or insufficient, provide a conservative description and preserve the uncertainty.
5. Use only the provided evidence and do not invent nodes, edges, paths, or communities.

User Prompt: Generate a local substructure description from the following evidence.
Target node: v_i
Node degree: d_i
Sampled neighbor count: n_i
Branching score: ρ_i^{tree}
Cyclic score: ρ_i^{cycle}
Hierarchical score: ρ_i^{hier}
Target-node text: t_i
Sampled neighbor texts: $\mathcal{T}_{\mathcal{N}(v_i)}$
First summarize the local connectivity and its main structural tendency, then use the textual evidence to summarize the local semantic context. Rely strictly on the provided evidence and avoid introducing unsupported structural facts.

Example:
Input: $d_i = 12$, $n_i = 4$, $\rho_i^{\text{tree}} = 0.78$, $\rho_i^{\text{cycle}} = 0.16$, and $\rho_i^{\text{hier}} = 0.42$; the target text concerns robust graph representation learning under noisy supervision; sampled neighbor texts concern semi-supervised graph encoding, topology-aware message passing, robust learning from noisy labels, and local graph structure.
Output: The target node has a relatively large local neighborhood. Its structural scores indicate a predominantly branching local structure with limited neighborhood closure and moderate degree asymmetry. The neighboring texts form a coherent semantic context related to graph representation learning and robustness under noisy supervision. These observations are limited to the provided local evidence.

The node-level input separates structural statistics, semantic context, and evidence limitations, reducing formatting variation and providing a stable input for the subsequent text encoder.

E.3 Generation and Encoding

We generate each node's local description q_i offline with Qwen3-8B using the prompt above. A frozen sentence-level text encoder then maps q_i to $\boldsymbol{c}_i$, as defined in Eq. 7. These representations provide the input to curvature routing during training.

F Experimental Details

F.1 Datasets

We evaluate R-MESD on five benchmark citation datasets. In each graph, nodes represent academic papers, edges represent citation relationships, and node labels correspond to paper categories. We use the following datasets:

- **Cora** (Sen et al., 2008): A benchmark citation network of scientific publications, where nodes are papers and labels indicate one of seven research topics.
- **Citeseer** (Giles et al., 1998): A citation network of scientific publications, in which each paper is assigned to one of six topic-based categories.
- **PubMed** (Sen et al., 2008): A biomedical citation network whose papers are connected by citations and grouped into three disease-related research categories.
- **ArXiv-2023** (He et al., 2024): A large-scale citation graph of arXiv papers covering 40 subject categories and diverse research areas.
- **OGBN-ArXiv** (Hu et al., 2020): A large-scale arXiv citation benchmark with 40 subject classes, designed to evaluate node representation learning at scale.

Detailed dataset statistics are provided in Table 5.

Table 5: Dataset statistics.

Datasets	#Classes	#Nodes	#Edges
Cora	7	2,708	5,429
Citeseer	6	3,186	4,277
PubMed	3	19,717	44,338
ArXiv-2023	40	46,198	78,543
OGBN-ArXiv	40	169,343	1,166,243

F.2 Baselines

We compare R-MESD with traditional graph learning methods and recent methods designed for text-attributed graphs. Traditional methods include the following:

- **MLP**: MLP predicts node labels from node features through fully connected layers, providing a semantic-only reference without aggregating graph neighbors.
- **GCN** (Kipf & Welling, 2017): GCN updates node representations through normalized neighborhood aggregation, allowing each node to incorporate information from directly connected neighbors.
- **GAT** (Velickovic et al., 2018): GAT learns attention weights for neighboring nodes and uses them to assign different importance to messages during graph convolution.
- **GraphSAGE** (Hamilton et al., 2017): GraphSAGE samples local neighborhoods and aggregates their features, enabling inductive node representation learning from a bounded amount of structural context.
- **GCNII** (Chen et al., 2020): GCNII improves deep graph convolution with initial residual connections and identity mapping, which help preserve input features and alleviate representation degradation.

Recent works designed for text-attributed graphs include:

- **SimTEG** (Duan et al., 2023): SimTEG combines textual attributes with graph structure to learn text-enhanced node representations for downstream graph prediction.

- **CGT** (Vakil & Amiri, 2024): CGT applies a graph transformer architecture to model long-range interactions between textual node attributes and graph neighborhoods.
- **G2P2** (Wen & Fang, 2023): G2P2 introduces graph prompt learning to connect textual knowledge with graph representations and improve transfer across graph tasks.
- **GLEM** (Zhao et al., 2023): GLEM jointly models language and graph representations through an iterative learning scheme that exchanges information between the two modules.
- **SLGNN** (Zhang et al., 2026): SLGNN incorporates structural information into graph neural representation learning to improve the modeling of topology in text-attributed graphs.
- **DeloopSGNN** (Li et al., 2026): DeloopSGNN reduces the influence of redundant local loops during propagation, aiming to preserve useful neighborhood signals in graph convolution.
- **AdaSpec** (Liu & Wang, 2026): AdaSpec uses adaptive spectral processing to transform graph signals and capture dataset-specific structural patterns.
- **LLM4NG** (Yu et al., 2025): LLM4NG employs large language models to enhance node representation learning by exploiting the semantic information contained in node text.
- **AD-GCN** (Hevapathige et al., 2026): AD-GCN adaptively adjusts graph convolution for text-attributed graph learning, allowing propagation to respond to node-level structural and semantic information.
- **RADE** (Saber & Salehi-Abari, 2026): RADE jointly models structural and semantic information with adaptive representation learning to reduce the limitations of fixed graph propagation.
- **TANS** (Wang et al., 2025d): TANS learns text-aware node representations with topology-adaptive neighborhood modeling, integrating textual attributes with locally relevant graph information.

F.3 Implementation Details

For our proposed R-MESD, we first pretrain the encoder and then use a GNN for node classification. We use Qwen3-8B to extract a local structural description for each node. Each training batch contains 32 target nodes, with at most 20 and 10 neighbors sampled at the first and second layers, respectively. Each branch contains three experts, with curvatures $\mathcal{K}_H = \{-0.1, -0.5, -1\}$ and $\mathcal{K}_S = \{0.1, 0.5, 1\}$ for the Hyperbolic and Spherical branches, respectively. The two branches are transformed to reference curvatures -1 and 1, respectively. We set the loss-balancing weights to $\lambda_{\text{str}} = 0.5$ and $\lambda_{\text{sem}} = 0.5$ and pretrain the encoder with a learning rate of 5×10^{-4}. We then freeze the encoder and concatenate its four 64-dimensional representations into a 256-dimensional input for the GNN classifier.

G Analysis and Statistics of Structural-Semantic Mismatch

Here we provide deeper empirical insights and rigorous quantitative evidence to justify our core motivation regarding the prevalence of structural-semantic mismatch in TAGs. While standard GNNs inherently assume topological-semantic homophily, we demonstrate both statistically and taxonomically that this underlying assumption is frequently violated across standard datasets.

G.1 Quantitative Distributional Divergence

To quantitatively assess the discrepancy between graph topology and textual semantics, we compute two key metrics across five datasets: (1) **Semantic Similarity**, measured via the cosine similarity of neighbor node embeddings encoded by Sentence-BERT, and (2) **Jensen-Shannon Divergence (JSD)**, which quantifies the statistical divergence between the neighborhood topological distributions and the text-derived semantic clusters.

As illustrated in Table 6, neighboring nodes across all datasets exhibit relatively low semantic similarity, with PubMed exhibiting the lowest score (0.356). Concurrently, the JSD values consistently exceed 0.400, reaching a peak of 0.727 on PubMed. These statistics provide robust empirical evidence that interconnected nodes in TAGs fail to necessarily share well-aligned semantic spaces, thereby substantiating the necessity of R-MESD's structural-semantic decoupling and adaptive Riemannian modeling.

Table 6: Quantitative statistics of structural-semantic mismatch on five datasets.

Metrics	Cora	Citeseer	PubMed	ArXiv-2023	OGBN-ArXiv
Semantic Similarity	0.480	0.563	0.356	0.546	0.454
Jensen-Shannon Divergence (JSD)	0.511	0.509	0.727	0.557	0.419

G.2 A Taxonomical Typology of Mismatched Edges

Through a granular review of the connected node pairs, we categorize the observed structural-semantic mismatch into four primary types, heavily drawing upon our empirical observations and verified statistics:

- **Different Topics:** In broad datasets such as OGBN-ArXiv and ArXiv-2023, cross-field connections frequently occur. As noted in our previous investigation, a computer vision paper may cite an NLP paper to reuse a model or methodology. Although a structural edge exists, their textual semantics may differ substantially. Quantitatively, this type accounts for 26.86% of connected edges in OGBN-ArXiv and 24.40% in ArXiv-2023.
- **Granular Focus: The Same Topic but Different Aspects:** In datasets like Cora and Citeseer, connected nodes often share a high-level research area but differ significantly in their specific focus. For instance, two papers in Cora may both belong to "Reinforcement Learning", yet one focuses on theoretical analysis while the other studies robotic control applications. This type accounts for 23.59% of connected edges in Cora and 19.57% in Citeseer.
- **Semantic Mismatch in Specialized Domains:** In domain-specific datasets like PubMed, papers often rely on tools, evidence, or findings from different subtopics. As a result, structurally connected papers may still exhibit substantial semantic differences. In PubMed, 35.17% of connected edges fall into the different topics category, while 39.90% correspond to the same topic but different aspects category, indicating that semantic mismatch is also prevalent in specialized domains.
- **Direct Contradiction or Inconsistencies:** Across all datasets, a structural edge does not necessarily imply agreement between two papers. For instance, a recent work may cite a previous method primarily to highlight its limitations. If a graph model directly aggregates features from such semantically inconsistent nodes, it may introduce harmful noise. Our analysis using Natural Language Inference (NLI) to detect textual contradictions further supports this observation, identifying potentially contradictory or inconsistent edges in 26.17% of connected pairs in Cora, 23.98% in Citeseer, 39.10% in PubMed, 33.60% in OGBN-ArXiv, and 33.92% in ArXiv-2023.

These four types highlight the complex interplay between graph structure and textual semantics in real-world graphs. Our proposed R-MESD is designed to effectively address such structural-semantic mismatch.

H Branch-Specific Geodesic Distances

To measure structural separation in each geometric branch, we use the geodesic distance induced by the κ-stereographic model. For a pair of nodes v_i and v_j, let $\xi \in \{H, S\}$ index the Hyperbolic and Spherical branches, respectively. The branch-specific distance is defined as:

$$d_{ij}^{\xi} = \frac{2}{\sqrt{|\kappa_\xi|}} \tan_{\kappa_\xi}^{-1}\left(\sqrt{|\kappa_\xi|}\,\| -\boldsymbol{p}_i^{\xi} \oplus_{\kappa_\xi} \boldsymbol{p}_j^{\xi}\|_2\right), \qquad \xi \in \{H, S\}. \tag{53}$$

Here, κ_ξ denotes the curvature associated with branch ξ. The vectors $\boldsymbol{p}_i^{\xi}$ and $\boldsymbol{p}_j^{\xi}$ are the structural positions of v_i and v_j in that branch, and d_{ij}^{ξ} is their geodesic distance. The quantity $\left\|-\boldsymbol{p}_i^{\xi} \oplus_{\kappa_\xi} \boldsymbol{p}_j^{\xi}\right\|_2$ measures the curvature-aware relative displacement between the two structural positions in model coordinates. The curvature-dependent inverse tangent and scale factor then convert this displacement magnitude into the length of the corresponding shortest geodesic.

Consequently, the same expression accommodates both branch geometries while retaining their different curvature properties. In the zero-curvature limit, it continuously reduces to the Euclidean distance.

The two branch-specific distances are then combined as:

$$d_{\mathrm{geo}}(i,j) = \sqrt{\left(d_{ij}^{H}\right)^2 + \left(d_{ij}^{S}\right)^2}. \tag{54}$$

This combined distance integrates structural information from both branches. It provides the geometric target for the structural consistency objective.