GRAPH MANIFOLD: RETHINKING GRAPH LEARNING THROUGH RIEMANNIAN GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) have achieved remarkable success in graph representation learning by modelling graph topology and the relationships among nodes embedded within it. However, existing graph learning typically regards graphs as discrete structures and optimises them based on local neighbourhood relationships, ignoring the intrinsic geometric properties and global evolution patterns. To address this, we rethink graph learning from the perspective of Riemannian geometry and introduce a novel **Graph Manifold**, which provides a continuous geometric space for graph representation and optimisation. Specifically, we reconstruct graph topology geometrically based on the Cholesky manifold, representing graphs as structured objects in a manifold space and enabling holistic geometric graph optimisation. Furthermore, we define a distributional structure on the Graph Manifold by modelling directional distributions in the tangent space, allowing the characterisation of intrinsic graph variations. Moreover, to extend the applicability of Graph Manifold to non-Euclidean computational spaces, we formulate Graph Manifold within a gyrovector algebraic framework and rigorously prove the intrinsic consistency of its geometric operations from a mathematical perspective. Finally, we apply Graph Manifold to the graph topology-oriented prompting task for pre-trained GNNs and conduct extensive experiments on four graph datasets under four pre-training strategies. The experimental results demonstrate that the proposed Graph Manifold consistently improves performance across multiple node classification tasks, validating its effectiveness as a novel Riemannian geometry for graph learning.

1 INTRODUCTION

Graph structures have become a fundamental data representation for modelling complex relationships and interactions, with widespread applications in social networks (Wei et al., 2023; Zhou et al., 2023), knowledge graphs (Wang et al., 2024), traffic networks (Qi et al., 2022), and healthcare (Cui et al., 2020; Fu et al., 2023). At the core of these applications lies graph topology, which encodes the structural relationships and interactions among entities and provides essential information for graph representation learning. In recent years, Graph Neural Networks (GNNs) (Chen et al., 2020a; Hamilton et al., 2017; Kipf & Welling, 2016a; Rossi et al., 2020) have gradually become a mainstream approach for graph representation learning by aggregating information interactions among nodes, and have achieved remarkable success in various graph-related tasks, including node classification (Luan et al., 2023), link prediction (Hwang et al., 2022), and graph classification (Li et al., 2022). However, despite the effectiveness of GNNs, graph topology is still primarily represented through discrete adjacency relationships between nodes, while existing graph learning methods mainly optimise graph representations based on local connections between nodes. Such a modelling paradigm overlooks the intrinsic geometric properties of graphs as holistic structures, as illustrated in Fig. 1.

To alleviate the above limitations, recent studies have enhanced graph representation capabilities from the perspectives of graph structure optimisation and graph structure fine-tuning. Pro-GNN (Jin et al., 2020) and IDGL (Chen et al., 2020b) jointly optimise graph structures and node representations by learning adaptive adjacency matrices or edge weights, thereby suppressing noisy connections in the original graphs and discovering potential topological relationships. Furthermore, with the emergence of graph pre-training paradigms, methods such as GraphPrompt (Liu et al., 2023) and

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.

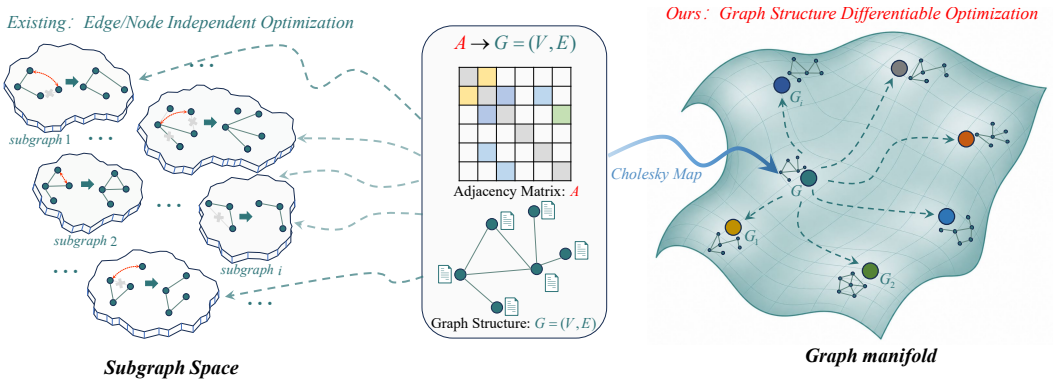


Figure 1: **From Discrete Graph Structures to Differentiable Graph Evolution.** Existing graph learning methods typically optimise discrete graph structures through local edge updates in the subgraph space. In contrast, we map the adjacency matrix to a point on the Graph Manifold, placing the entire graph structure in a continuous Riemannian space rather than assigning spatial positions to individual nodes, thereby enabling differentiable graph evolution.

GraphTOP (Fu et al., 2026) introduce topology refinement strategies to optimise original graphs by leveraging structural priors learned from large-scale graph data, thus obtaining more effective graph structures that better align with underlying data distributions. However, despite their effectiveness in improving graph structure representation, these methods still rely on the assumption of a discrete graph space. They represent graphs as deterministic objects defined by adjacency relationships and obtain a single structural representation through local edge-wise optimisation, making them difficult to capture the intrinsic uncertainty inherent in graph structures.

To characterise this uncertainty, many studies have explored more robust graph representation methods from the perspective of distribution learning. VGAE (Kipf & Welling, 2016b) introduces latent variable distributions to model graph structures probabilistically, enabling the learning of potential structural relationships. Furthermore, Bayesian GNNs (Zhang et al., 2019) probabilistically model uncertainty in model parameters or node representations, improving adaptability to unknown variations and noisy data. In addition, methods such as UGSL (Fatemi et al., 2023) dynamically adjust graph structures by estimating the uncertainty of edge relationships, thereby obtaining more robust topological representations. However, although the aforementioned methods introduce probabilistic modelling to characterise uncertainty in graph structures, their modelling aims are primarily limited to capturing the probabilistic variations of local edge relationships or latent node representations, rather than directly modeling the distribution of the graph structure itself. Therefore, it is natural to pose the question: *Can we construct a continuous graph structure space and further characterise the intrinsic distribution patterns of graph structures within it?*

To answer this question, we provide the first investigation of graph learning from the perspective of Riemannian geometry and propose a novel Graph Manifold, which aims to construct a continuous geometric space for graph structures to characterise their intrinsic properties. Existing graph methods primarily model and optimise graph structures through local operations on edges, such that local changes may affect the structure of the entire graph. Therefore, we treat the complete graph topology as a holistic geometric object and model it on a Riemannian manifold, enabling complex structural variations to be characterised within a low-dimensional geometric space and the globally constrained relationships between different graph topologies to be captured. Specifically, we geometrically represent graph topology based on the Cholesky manifold, encoding each complete graph topology as a structural matrix and mapping it to a point on the Riemannian manifold. Thus, each point on Graph Manifold corresponds to a complete graph topology rather than an individual node or edge. Furthermore, we model directional distributions of graph structures in the local tangent space and define the intrinsic distributional structure of Graph Manifold, thereby effectively capturing latent structural patterns and uncertainties underlying graph variations. In addition, we extend the geometric operations of Graph Manifold to a gyrovector algebraic framework, which provides algebraic expressions for geometric operations in non-Euclidean spaces, and theoretically prove that this algebraic framework can describe the Riemannian geometric operations on Graph Manifold.

We summarise the main contributions as follows:

- We provide the first investigation of graph learning from the perspective of Riemannian geometry and propose Graph Manifold, which reformulates graph structures as geometric objects in a continuous manifold space.
- We further define the intrinsic distributional representation of graph structures on Graph Manifold by modelling their latent variations and uncertainties.
- We extend the geometric operations of Graph Manifold to a gyrovector algebraic framework and theoretically prove its consistency with the underlying Riemannian structure.

2 Related Works

2.1 Graph Structure Learning

Graph structure learning aims to automatically discover latent relationships between nodes within a graph from data and enhance the representation capability of Graph Neural Networks (GNNs) (Xu et al., 2018) for complex relational patterns by optimising graph topologies. Given an observed graph structure, existing graph structure learning methods refine its topology by learning node relationships and correspondingly adding, removing, or reweighting edges. Existing graph structure learning methods optimise topology by learning node relationships and refining pre-trained graph structures. Methods such as GAS (Fey et al., 2021) and IDGL (Chen et al., 2020b) model node feature similarities or representation consistency to dynamically infer potential edges and update adjacency structures, thereby obtaining graph topologies that better align with the underlying data distribution. Methods such as Pro-GNN (Jin et al., 2020) and GRCN (Yu et al., 2020) further introduce structural prior constraints and leverage priors such as sparsity and smoothness to suppress noisy connections, improving the stability and reliability of the learned graph structures. Furthermore, with the development of graph pre-training, recent studies have explored topology refinement as a means of adapting pre-trained GNNs to downstream tasks. Methods such as GraphPrompt (Liu et al., 2023) and GraphTOP (Fu et al., 2026) adjust input graph structures through structural prompting, enabling pre-trained GNNs to better adapt to downstream tasks. Despite their effectiveness in improving graph structure representations, these methods are all built upon the assumption of a discrete graph space, where graphs are represented as discrete objects defined by adjacency matrices or edge relationships. To address this limitation, we propose Graph Manifold, which, for the first time, reformulates graph structures as geometric objects in a continuous Riemannian manifold space. By constructing continuous geometric representations of graph structures, Graph Manifold enables a transition from discrete topology optimisation to continuous geometric graph learning.

2.2 Graph Distribution Modelling

Graph distribution modelling aims to characterise the latent structural patterns, uncertainties, and variations of graph data from a probabilistic perspective. Existing studies explore graph distribution representations from the perspectives of latent distribution learning and uncertainty-aware representation learning. Variational learning-based methods (*e.g.*, VGAE (Kipf & Welling, 2016b) and ARVGA (Pan et al., 2018)) introduce latent variable spaces and employ variational inference to learn distributions over node representations, thereby discovering potential connections beyond observed relationships. With the development of generative modelling, methods such as GraphRNN (You et al., 2018), GraphVAE (Simonovsky & Komodakis, 2018), GraphAF (Shi et al., 2020), and GraphDF (Luo et al., 2021) learn graph structural distributions by modelling the discrete generation process of nodes and their connectivity. Furthermore, Bayesian graph learning methods (e.g., Bayesian GCN and Bayesian GNN (Zhang et al., 2019)) model the parameters probabilistically to characterise parameter uncertainty, thereby enhancing robustness to noisy data and distribution shifts. In addition, methods such as UGSL (Fatemi et al., 2023) incorporate uncertainty estimation into graph structure learning and dynamically adjust edge connection probabilities to obtain more reliable topological representations. Although these methods explore probabilistic representations of graphs from different perspectives, their distribution modelling objects tend to focus on local edge relationships, rather than the intrinsic distributional characteristics of graph structures. To address this limitation, based on the proposed Graph Manifold, we further define an intrinsic distributional representation of graph structures, which directly characterises structural variations and

uncertainties within a continuous Riemannian geometric space, achieving a transition from conventional probabilistic graph modelling to intrinsic graph distribution modelling.

3 PRELIMINARIES

3.1 CHOLESKY MANIFOLD

The Cholesky manifold provides a continuous Riemannian space for representing lower-triangular matrices with strictly positive diagonal entries (Chen et al., 2026). It is defined as

$$\mathcal{L}_+^n = \left\{ L \in \mathbb{R}^{n \times n} \mid L_{ij} = 0,\ i < j,\ L_{ii} > 0 \right\}. \tag{1}$$

By equipping this space with an appropriate Riemannian metric, matrix structures can be compared and optimised according to their intrinsic geometry rather than the conventional Euclidean structure (Lin, 2019). In our framework, we exploit this geometric structure to model graph-related matrix representations in a continuous space. The tangent space, Riemannian metric, exponential and logarithmic maps, and geodesic distance used throughout this work are summarised in Appendix D.1.

3.2 GYROVECTOR SPACES

To characterise the non-Euclidean algebraic structure of the proposed Graph Manifold, we further establish a gyrovector framework based on its intrinsic Riemannian geometry (Ungar, 2002; 2005; 2014). Specifically, we define the corresponding gyro addition, gyration, and scalar multiplication on $\mathcal{A}^n$ using the exponential map, logarithmic map, and parallel transport, and establish the resulting gyrovector structure. The formal definitions, algebraic operations, and complete axiomatic verification are provided in Appendix D.2.

4 GRAPH MANIFOLD

Graph structures are inherently discrete objects, whose topological relationships are typically represented by adjacency matrices. To characterise variations in graph topology within a continuous space, we introduce Riemannian geometry to transform discrete graph structures into continuous geometric representations and, construct a graph manifold.

4.1 GRAPH MANIFOLD CONSTRUCTION

Graph structures are inherently discrete objects, whose topological relationships are typically represented by adjacency matrices. For an undirected graph $G = (V, E, A)$ with n nodes, where V and E denote the node and edge sets, respectively, $A \in \mathbb{R}^{n \times n}$ represents the corresponding adjacency matrix (Zhang et al., 2020). Due to the symmetric connectivity between nodes in an undirected graph, its adjacency matrix satisfies $A = A^\top$.

Different graph structures are typically represented by discrete adjacency matrices, with topology variations manifested as discrete adjustments to edge connectivity. During topology optimisation, edges are typically adjusted independently based on local node-pair relationships, without explicitly modelling structural dependencies between edges or global coordination. Consequently, this paradigm makes it difficult to characterise the global structural relationships and coordinated variations of complete graph topologies. To address this issue, we aim to transform graph topology from mutually independent discrete structures into continuously controllable structural representations. Since the upper-triangular and lower-triangular parts of a symmetric adjacency matrix contain redundant topological information, we retain only one side of the independent structural information and reparameterise it as a lower-triangular matrix (Golub & Van Loan, 2013):

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}, \qquad l_{ii} > 0. \tag{2}$$

Although the adjacency matrix typically has a zero diagonal, GNN message passing commonly retains self-information. Therefore, we constrain the diagonal entries of L to be strictly positive, where l_{ii} represents the self-information contribution of node i. This design both satisfies the positived iagonal constraint of the Cholesky manifold and preserves node self-information. The lower-triangular entries of L are constrained to $[0, 1]$ and encode the connectivity strength between node pairs. Thus, each edge can be continuously strengthened or weakened through its corresponding entry, providing a continuous representation of graph topology. According to the Cholesky manifold $\mathcal{L}_+^n$ defined previously (see Eq. 1, L can be regarded as a point on this manifold, thereby transforming the originally mutually independent discrete graph structures into continuous structural representations. Furthermore, to establish a correspondence between this continuous representation and the actual graph topology, we define a structural mapping from the Cholesky manifold to the adjacency: $\Phi : \mathcal{L}_+^n \to \mathbb{R}^{n\times n}$, and map L to its corresponding adjacency matrix as

$$A(L) = \Phi(L) = L + L^\top - \mathrm{diag}(L). \tag{3}$$

This mapping exploits the Cholesky information to recover the complete topology of the undirected graph. Accordingly, a correspondence between the continuous structural representation on the Cholesky manifold and the corresponding graph topology is established as

$$L \in \mathcal{L}_+^n \xrightarrow{\Phi} A(L) \in \mathcal{A}^n \longleftrightarrow G. \tag{4}$$

Here, $\mathcal{A}^n$ denotes the proposed Graph Manifold, *i.e.*, the space of graph topologies defined over (n) nodes, where each point $A(L) \in \mathcal{A}^n$ corresponds to a complete graph topology G. Thus, as L continuously varies on the Cholesky manifold $\mathcal{L}_+^n$, the corresponding adjacency matrix $A(L)$ undergoes corresponding structural variations, thereby transforming the discrete topological adjustments of conventional graph structures into continuous structural evolution on the manifold. Based on this continuous representation, we further introduce a Riemannian geometric structure on $\mathcal{L}_+^n$ to characterise the intrinsic geometric relationships between different graph topologies.

4.2 Riemannian Structure of Graph Manifold

In this subsection, for $L \in \mathcal{L}_+^n$, we incorporate the structural mapping Φ into the tangent space (Boumal, 2020), Riemannian metric, geodesic, exponential map, logarithmic map, and geodesic distance, thereby establishing a unified geometric description of graph-topological variations.

Proposition 1 (Tangent Space of the Graph Manifold)**.** *For any $L \in \mathcal{L}_+^n$, the structural mapping Φ maps a tangent vector $X \in T_L\mathcal{L}_+^n$ on the Cholesky manifold to a tangent vector on the Graph Manifold $\mathcal{A}^n$ as $\Delta A_X = \mathrm{d}\Phi_L[X] = X + X^\top - \mathrm{diag}(X)$. Accordingly, the tangent space of the Graph Manifold at $A(L) = \Phi(L)$ is given by*

$$T_{A(L)}\mathcal{A}^n = \left\{\mathrm{d}\Phi_L[X] \mid X \in T_L\mathcal{L}_+^n\right\}, \tag{5}$$

where every tangent vector ΔA_X is symmetric.

Proposition 2 (Riemannian Metric of the Graph Manifold)**.** *For any $L \in \mathcal{L}_+^n$ and any two tangent vectors $\Delta A_X, \Delta A_Y \in T_{A(L)}\mathcal{A}^n$, the Riemannian metric of the Graph Manifold $\mathcal{A}^n$ at $A(L)$ is defined as*

$$g^{\mathcal{A}}_{A(L)}(\Delta A_X, \Delta A_Y) = \sum_{i>j}(\Delta A_X)_{ij}(\Delta A_Y)_{ij} + \sum_{j=1}^{n}\frac{(\Delta A_X)_{jj}(\Delta A_Y)_{jj}}{L_{jj}^2}. \tag{6}$$

Proposition 3 (Geodesic on the Graph Manifold)**.** *For any $L \in \mathcal{L}_+^n$ and any tangent vector $\Delta A_X \in T_{A(L)}\mathcal{A}^n$, let $X \in T_L\mathcal{L}_+^n$ satisfy $\Delta A_X = X + X^\top - \mathrm{diag}(X)$. Then, the geodesic on the Graph Manifold emanating from $A(L)$ along the initial direction ΔA_X is defined as $\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t) = \Phi\left(\widetilde{\gamma}_{L,X}(t)\right)$, $\widetilde{\gamma}_{L,X}(t)$ denotes the geodesic on the Cholesky manifold emanating from L along the initial direction X, given by*

$$\widetilde{\gamma}_{L,X}(t) = [L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right), \tag{7}$$

Proposition 4 (Riemannian Exponential Map on the Graph Manifold)**.** *For any $L \in \mathcal{L}_+^n$ and any $\Delta A_X \in T_{A(L)}\mathcal{A}^n$, let $X \in T_L\mathcal{L}_+^n$ satisfy $\Delta A_X = X + X^\top - \mathrm{diag}(X)$. Then, the Riemannian exponential map on the Graph Manifold is defined as*

$$\mathrm{Exp}^{\mathcal{A}}_{A(L)}(\Delta A_X) = \Phi\left(\widetilde{\mathrm{Exp}}_L(X)\right), \tag{8}$$

where, $\widetilde{\mathrm{Exp}}_L(X) = [L] + [X] + D(L)\exp\left(D(X)D(L)^{-1}\right)$.

Proposition 5 (Riemannian Logarithmic Map on the Graph Manifold)**.** *For any two points* $A(L), A(K) \in \mathcal{A}^n$ *on the Graph Manifold, where* $L, K \in \mathcal{L}_+^n$*, the Riemannian logarithmic map with* $A(L)$ *as the base point is defined as*

$$\mathrm{Log}^{\mathcal{A}}_{A(L)}(A(K)) = \Delta A_{L\to K} \in T_{A(L)}\mathcal{A}^n, \tag{9}$$

where, $\Delta A_{L\to K} = X_{L\to K} + X_{L\to K}^\top - \mathrm{diag}(X_{L\to K})$*, and* $X_{L\to K}$ *denotes the corresponding tangent vector on the Cholesky manifold, given by*

$$X_{L\to K} = [K] - [L] + D(L)\log\left(D(L)^{-1}D(K)\right). \tag{10}$$

4.3 Intrinsic Graph Distribution

In graph structure learning, representing a graph by a single deterministic adjacency matrix can only describe a particular topological state, making it difficult to characterise the inherent uncertainty arising from local topological variations. Based on the Graph Manifold constructed above, we extend the probabilistic representation of graph structures from the adjacency space to its intrinsic geometric space.

For a reference graph structure $A(L) \in \mathcal{A}^n$ on the Graph Manifold, the corresponding tangent space is $T_{A(L)}\mathcal{A}^n$. For any underlying Cholesky variation direction X, its corresponding Graph Manifold tangent vector is defined as

$$\Delta A_X = X + X^\top - \mathrm{diag}(X). \tag{11}$$

Thus, ΔA_X represents an infinitesimal change in graph topology along a local direction with $A(L)$ as the reference point. ΔA_X characterises the local variation of the graph structure relative to its current topological state, and can therefore serve as the fundamental geometric variable for modelling graph topological uncertainty.

Based on this observation, we model local topological variations probabilistically in the tangent space of the Graph Manifold. Specifically, at $A(L)$, we define the probability distribution of local graph-topological variations as $q_{A(L)}(\Delta A_X), \Delta A_X \in T_{A(L)}\mathcal{A}^n$. We further parameterise this local variation distribution as

$$q_{A(L)}(\Delta A_X) = \mathcal{N}\left(\Delta A_X; \mu_{\mathcal{A}}, \Sigma_{\mathcal{A}}\right). \tag{12}$$

where $\mu_{\mathcal{A}}$ describes the mean topological variation direction from the current graph structure $A(L)$, while $\Sigma_{\mathcal{A}}$ characterises the variation across different local topological directions, thereby capturing the intrinsic uncertainty of the graph structure in the local topological space.

The above probability distribution is still defined in the tangent space $T_{A(L)}\mathcal{A}^n$, where the random variable ΔA_X represents a local topological variation rather than an actual graph structure on the Graph Manifold. To transform the local variation in the tangent space into a graph structure on the Graph Manifold, we employ the Riemannian exponential map defined above:

$$A' = \mathrm{Exp}^{\mathcal{A}}_{A(L)}\left(\Delta A_X\right). \tag{13}$$

Here, A' denotes a particular graph structure sample induced by the random local topological variation ΔA_X. Consequently, the random local topological variation in the tangent space is transformed into a random graph structure on the Graph Manifold through the Riemannian exponential map. Based on this construction, we further define the Intrinsic Graph Distribution on the Graph Manifold as

$$A \sim \widetilde{\mathcal{N}}_{\mathcal{A}}\left(\mathcal{A}^n; \mu_{\mathcal{A}}, \Sigma_{\mathcal{A}}\right), \tag{14}$$

where $\widetilde{\mathcal{N}}_{\mathcal{A}}$ denotes the intrinsic Gaussian distribution induced by the Riemannian geometry of the Graph Manifold.

5 Gyrovector Extension of Graph Manifold

The Graph Manifold $\mathcal{A}^n$ established above provides a Riemannian geometric space for continuously characterising graph-topological variations. Following (Nguyen (2022); Ganea et al. (2018); Skopek

et al. (2019); Nguyen & Yang (2023)), a Riemannian manifold equipped with exponential maps, logarithmic maps, and parallel transport can be endowed with a corresponding gyro structure. We therefore instantiate this construction directly on the Graph Manifold $\mathcal{A}^n$.

Let E denote a fixed reference graph structure on the Graph Manifold, which serves as the gyro identity element. Specifically, let L_E be the Cholesky representation associated with the reference graph, and define

$$E = A(L_E) = L_E + L_E^\top - \mathrm{diag}(L_E). \tag{15}$$

For any graph structures $A(L), A(K), A(H) \in \mathcal{A}^n$, the corresponding gyro operations are defined using the intrinsic Riemannian geometry of the Graph Manifold.

Lemma 1 (Gyro Addition)**.** *Let $A(L), A(K) \in \mathcal{A}^n$ and let $E = A(L_E)$ be the reference graph structure. Under the Riemannian gyro addition defined by the exponential map, logarithmic map, and parallel transport on $\mathcal{A}^n$,*

$$A(L) \oplus_{\mathcal{A}} A(K) = \mathrm{Exp}^{\mathcal{A}}_{A(L)} \left(\mathcal{PT}^{\mathcal{A}}_{E \to A(L)} \left(\mathrm{Log}^{\mathcal{A}}_{E}(A(K)) \right) \right), \tag{16}$$

the resulting graph structure can be expressed as $A(L) \oplus_{\mathcal{A}} A(K) = A(L_\oplus)$, Equivalently,

$$A(L) \oplus_{\mathcal{A}} A(K) = [L] + [K] - [L_E] + \left([L] + [K] - [L_E] \right)^\top + D(L)D(L_E)^{-1}D(K). \tag{17}$$

Lemma 2 (Gyro Inverse)**.** *Let $E = A(L_E)$ denote the identity element of the Graph Manifold. For any $A(L) \in \mathcal{A}^n$, its gyro inverse $\ominus_{\mathcal{A}} A(L)$ is given by*

$$\ominus_{\mathcal{A}} A(L) = A\left(L_{\mathcal{A}}^{-1}\right), \tag{18}$$

where $L_{\mathcal{A}}^{-1} = 2[L_E] - [L] + D(L_E)D(L)^{-1}D(L_E)$. Equivalently, its strict lower-triangular and diagonal components are $[L_{\mathcal{A}}^{-1}] = 2[L_E] - [L]$, and $D(L_{\mathcal{A}}^{-1}) = D(L_E)D(L)^{-1}D(L_E)$.

Lemma 3 (Gyro Scalar Multiplication)**.** *Let $A(L) \in \mathcal{A}^n$, let $t \in \mathbb{R}$, and let $E = A(L_E)$ be the reference graph structure. The gyro scalar multiplication induced by the Riemannian exponential and logarithmic maps is given by*

$$t \odot_{\mathcal{A}} A(L) = \mathrm{Exp}^{\mathcal{A}}_{E} \left(t \, \mathrm{Log}^{\mathcal{A}}_{E}(A(L)) \right). \tag{19}$$

Equivalently, in the graph space,

$$t \odot_{\mathcal{A}} A(L) = [L_E] + t([L] - [L_E]) + \left([L_E] + t([L] - [L_E]) \right)^\top + D(L_E) \left(D(L_E)^{-1} D(L) \right)^t. \tag{20}$$

Definition 1 (Graph Gyrovector Structure)**.** *For any $A(L), A(K) \in \mathcal{A}^n$, define the gyroautomorphism generated by $A(L)$ and $A(K)$ as the identity mapping*

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{Id}_{\mathcal{A}}. \tag{21}$$

Theorem 1 (Graph Gyrovector Space)**.** *Let $\mathcal{A}^n$ be the Graph Manifold equipped with the gyro addition $\oplus_{\mathcal{A}}$ and the gyro scalar multiplication $\odot_{\mathcal{A}}$ defined above. Then $(\mathcal{A}^n, \oplus_{\mathcal{A}}, \odot_{\mathcal{A}})$ forms a Graph Gyrovector Space.*

6 Experiments

To validate the effectiveness of the proposed Graph Manifold and its distribution modelling approach, we conduct experiments on graph fine-tuning tasks. We focus on evaluating whether graph representations that incorporate intrinsic geometric structures and structural uncertainty can improve the performance of downstream graph learning tasks.

6.1 Experimental Setup

Datasets. We select four real-world graph datasets from diverse domains to comprehensively evaluate the effectiveness of the proposed framework. Specifically, we use Cora (Yang et al., 2016), PubMed (Yang et al., 2016), Amazon (Platonov et al., 2023), and Minesweeper (Platonov et al., 2023). Further details are provided in Appendix G.1.

Pre-training Strategies. To assess the adaptability of the proposed framework to different pre-training paradigms, we select four representative pre-training strategies for evaluation. For contrastive learning-based methods, we adopt GraphCL (You et al., 2020) and SimGRACE (Xia et al., 2022); for generative methods, we follow the existing studies GPPT (Sun et al., 2022) and GraphPrompt (Liu et al., 2023). Further details are provided in Appendix G.2and Appendix G.3.

6.2 Evaluation of Effectiveness

We first evaluate the overall performance of GraphManifold* against existing graph prompting methods under different pre-training strategies. Table 1 reports the 5-shot node classification mean accuracy across four datasets under four pre-training strategies. Overall, GraphManifold* demonstrates stable and competitive performance across different pre-training strategies and datasets, achieving the best results in 13 out of 16 experimental settings and the second-best results in 2 settings. Notably, GraphManifold* maintains a consistent advantage across all four pre-training strategies. Under GraphCL and LP-GraphPrompt, GraphManifold* achieves the best results on all four datasets. Under SimGRACE, it achieves the best results on Cora, PubMed, and Minesweeper, while obtaining the second-best result on Amazon. Under LP-GPPT, it achieves the best results on Cora and PubMed and the second-best result on Minesweeper. These results demonstrate that elevating graph topology from discrete local connectivity patterns to a global structure in a continuous geometric space, as well as modelling the intrinsic distribution of graph structures within this space, provides a more natural way to characterise the overall evolution of graph topology during structural changes.

Table 1: Performance comparison of different graph prompting methods under different pre-training strategies. **Bold** and underline denote the best and second-best results, respectively.

Pre-training	Graph Prompting	Cora	PubMed	Amazon	Minesweeper
GraphCL	Linear Probe	55.69%±5.74	67.30%±6.26	23.19%±7.21	67.59%±6.30
	GPPT	61.50%±4.49	65.75%±3.99	24.27%±3.74	65.44%±8.97
	ALL-in-one	52.33%±4.55	65.78%±8.65	22.82%±6.09	63.82%±8.63
	GraphPrompt	62.12%±3.28	67.01%±4.56	21.71%±2.93	61.19%±3.50
	GraphPrompt+	58.91%±3.12	66.26%±5.75	23.83%±2.30	61.64%±6.36
	ProNoG	60.01%±7.03	68.17%±4.82	23.26%±2.42	65.48%±3.40
	GraphTOP	63.44%±4.21	68.28%±4.15	23.52%±5.02	68.25%±7.14
	GraphManifold*	**64.13%±3.83**	**68.94%±4.21**	**24.41%±2.66**	**71.31%±6.76**
SimGRACE	Linear Probe	40.68%±2.29	54.59%±6.02	24.58%±4.18	60.58%±6.42
	GPPT	44.83%±4.67	52.25%±5.91	24.27%±3.74	59.62%±4.80
	ALL-in-one	41.11%±4.92	51.45%±4.73	22.66%±3.55	58.11%±3.82
	GraphPrompt	47.02%±3.87	55.74%±5.80	21.24%±2.78	58.72%±4.37
	GraphPrompt+	51.26%±4.90	55.93%±6.98	**25.07%±1.71**	60.76%±6.75
	ProNoG	42.44%±2.97	55.11%±5.98	22.53%±2.65	63.03%±2.74
	GraphTOP	50.57%±2.91	56.64%±5.42	23.32%±2.34	61.25%±5.08
	GraphManifold*	**59.01%±3.93**	**58.41%±6.21**	24.81%±2.07	**68.71%±6.14**
LP-GPPT	Linear Probe	24.40%±2.83	42.26%±5.09	25.50%±4.11	63.22%±8.16
	GPPT	32.08%±7.66	44.85%±4.73	**28.90%±3.50**	63.44%±8.28
	ALL-in-one	26.67%±6.24	41.11%±4.92	24.49%±3.51	59.97%±4.67
	GraphPrompt	30.14%±2.01	44.72%±6.68	21.88%±3.56	61.73%±6.35
	GraphPrompt+	33.42%±2.91	45.17%±6.91	24.34%±1.72	61.15%±3.37
	ProNoG	33.71%±4.12	46.07%±3.62	21.39%±1.69	**66.11%±4.20**
	GraphTOP	31.96%±2.20	44.52%±6.18	28.77%±6.84	61.48%±11.29
	GraphManifold*	**34.48%±3.81**	**46.31%±3.65**	28.20%±4.30	64.14%±3.10
LP-GraphPrompt	Linear Probe	50.15%±5.88	66.26%±5.69	25.00%±6.78	65.90%±7.36
	GPPT	52.13%±7.15	63.16%±8.25	25.38%±5.78	62.53%±8.91
	ALL-in-one	49.42%±2.70	64.73%±6.46	21.37%±3.65	58.17%±4.67
	GraphPrompt	52.35%±4.82	68.16%±8.23	22.76%±2.81	58.01%±3.26
	GraphPrompt+	52.19%±5.22	62.19%±6.70	24.44%±1.81	61.78%±3.92
	ProNoG	52.49%±5.43	67.68%±5.02	23.79%±2.04	59.88%±8.50
	GraphTOP	53.44%±5.02	68.14%±5.47	25.01%±4.51	67.09%±8.45
	GraphManifold*	**54.06%±5.32**	**70.26%±5.87**	**25.54%±4.90**	**69.11%±7.84**

6.3 Ablation Analysis

Graph Manifold We first investigate the effect of incorporating Riemannian manifold geometry by comparing GraphEuclidean with GraphManifold. Specifically, GraphEuclidean models graph structures in a Euclidean space, whereas GraphManifold represents graph topology on the contin-

uous graph manifold proposed in this work. As shown in Table 2, incorporating manifold geometry does not yield consistent performance improvements across all datasets, but still demonstrates clear advantages across multiple datasets and pre-training strategies. In particular, GraphManifold consistently outperforms GraphEuclidean on PubMed and Minesweeper under both GraphCL and SimGRACE, and also achieves better performance on these two datasets under LP-GPPT. These results suggest that modelling graph structures on a Riemannian manifold provides a more suitable geometric representation space for capturing variations in graph topology. However, relying solely on manifold geometry remains insufficient to consistently improve performance across all datasets.

Table 2: Ablation study on the effects of Riemannian manifold geometry and intrinsic distribution modelling across four pre-training strategies. **Bold** denotes the best performance within each group.

Pre-training	Methods	Manifold	Dist.	Cora	PubMed	Amazon	Minesweeper
GraphCL	GraphEuclidean	×	×	61.56%±4.77	67.68%±6.94	22.53%±3.86	70.40%±7.19
	GraphManifold	✓	×	59.96%±4.21	68.40%±6.41	22.70%±3.64	71.26%±7.32
	GraphManifold*	✓	✓	**64.13%±3.83**	**68.94%±4.21**	**24.41%±2.66**	**71.31%±6.76**
SimGRACE	GraphEuclidean	×	×	47.00%±3.50	57.38%±7.09	24.00%±2.41	67.36%±11.19
	GraphManifold	✓	×	44.80%±3.80	58.16%±6.73	24.33%±2.49	67.86%±10.54
	GraphManifold*	✓	✓	**59.01%±3.93**	**58.41%±6.21**	**24.81%±2.07**	**68.71%±6.14**
LP-GPPT	GraphEuclidean	×	×	31.51%±3.48	37.99%±4.19	25.97%±5.27	62.98%±7.42
	GraphManifold	✓	×	29.94%±3.97	38.64%±3.38	25.10%±5.69	63.39%±6.70
	GraphManifold*	✓	✓	**34.4%±3.81**	**46.31%±3.65**	**28.20%±4.30**	**64.14%±3.10**
LP-GraphPrompt	GraphEuclidean	×	×	53.70%±5.32	70.15%±5.87	25.23%±3.37	68.02%±7.75
	GraphManifold	✓	×	52.41%±5.72	69.39%±6.76	24.94%±4.94	68.29%±7.92
	GraphManifold*	✓	✓	**54.06%±5.32**	**70.26%±5.87**	**25.54%±4.90**	**69.11%±7.84**

Graph Distribution Modelling. We also investigate the contribution of intrinsic distribution modelling by comparing GraphManifold* with GraphManifold. The difference between the two variants is whether, in addition, GraphManifold* explicitly models the intrinsic distribution of graph structures in the tangent space of the graph manifold. As shown in Table 2, introducing distribution modelling leads to substantial and stable performance improvements across all four pre-training strategies. In particular, under LP-GPPT, distribution modelling yields improvements of 4.46%, 7.67%, 3.10%, and 0.75% on Cora, PubMed, Amazon, and Minesweeper, respectively. Similar improvements can also be observed under LP-GraphPrompt, where GraphManifold* consistently outperforms GraphManifold across all four datasets. These results demonstrate that modelling the intrinsic distribution of graph structures effectively complements the underlying manifold geometry, enabling the model to more adequately capture structural variations and uncertainty during graph topological changes. Overall, the ablation study verifies that the performance gains of GraphManifold* arise not only from representing graph topology in a Riemannian space, but also from further explicitly modelling the intrinsic distribution of graph structures within this space. This further suggests that the manifold provides the geometric space, while distribution modelling characterises how graph structures are intrinsically distributed within that space.

7 Conclusion

This paper addresses the challenge that existing graph learning methods struggle to characterise graph topological variations and their intrinsic uncertainty from a holistic perspective. From the viewpoint of Riemannian geometry, we propose Graph Manifold. We first construct a graph manifold to provide a unified and continuous geometric representation of graph topology, and further model the intrinsic distribution of graph structures within this space to characterise the underlying structural patterns and uncertainty arising from graph topological variations. Finally, we extend the geometric operations on Graph Manifold to a gyrovector space and theoretically prove their consistency with the underlying Riemannian geometric structure. Overall, this paper establishes a theoretical learning framework for Graph Manifold, providing a new perspective for characterising graph structures through continuous geometry.

ETHICS STATEMENT

This work is evaluated on publicly available benchmark datasets, including Cora, PubMed, Amazon, and Minesweeper, none of which contain personally identifiable or sensitive information. The study involves no human subjects or sensitive data, and we identify no ethical concerns associated with this work.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, the experimental section provides detailed information on the Cora, PubMed, Amazon, and Minesweeper datasets, experimental settings, and evaluation metrics, while the supplementary materials provide further implementation details, experimental configurations, and results. If the paper is accepted, we will release the complete source code and relevant configurations to support independent verification and reproduction of the reported results.

REFERENCES

Nicolas Boumal. An introduction to optimization on smooth manifolds. *To appear with Cambridge University Press,(Apr 2022)*, 2020.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International conference on machine learning*, pp. 1725–1735. PMLR, 2020a.

Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in neural information processing systems*, 33:19314–19326, 2020b.

Ziheng Chen, Yue Song, Xiaojun Wu, and Nicu Sebe. Fast and stable riemannian metrics on spd manifolds via cholesky product geometry. In *International Conference on Learning Representations*, volume 2026, pp. 136758–136786, 2026.

Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 492–502, 2020.

Bahare Fatemi, Sami Abu-El-Haija, Anton Tsitsulin, Mehran Kazemi, Dustin Zelle, Neslihan Bulut, Jonathan Halcrow, and Bryan Perozzi. Ugsl: A unified framework for benchmarking graph structure learning. *arXiv preprint arXiv:2308.10737*, 2023.

Matthias Fey, Jan E Lenssen, Frank Weichert, and Jure Leskovec. Gnnautoscale: Scalable and expressive graph neural networks via historical embeddings. In *International conference on machine learning*, pp. 3294–3304. PMLR, 2021.

Xingbo Fu, Chen Chen, Yushun Dong, Anil Vullikanti, Eili Klein, Gregory Madden, and Jundong Li. Spatial-temporal networks for antibiogram pattern prediction. In *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)*, pp. 225–234. IEEE, 2023.

Xingbo Fu, Zhenyu Lei, Zihan Chen, Binchi Zhang, Chuxu Zhang, and Jundong Li. Graphtop: Graph topology-oriented prompting for graph neural networks. *Advances in Neural Information Processing Systems*, 38:158309–158335, 2026.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

EunJeong Hwang, Veronika Thost, Shib Sankar Dasgupta, and Tengfei Ma. An analysis of virtual nodes in graph neural networks for link prediction. In *The first learning on graphs conference*, 2022.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Sejong Kim. Distributivity on the gyrovector spaces. *Kyungpook Math. J*, 55(1):13–20, 2015.

Sejong Kim. Gyrovector spaces on the open convex cone of positive definite matrices. *Mathematics Interdisciplinary Research*, 1(1):173–185, 2016.

Sejong Kim. Operator inequalities and gyrolines of the weighted geometric means. *arXiv preprint arXiv:2009.10274*, 2020a.

Sejong Kim. Ordered gyrovector spaces. *Symmetry*, 12(6):1041, 2020b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

Shuangli Li, Jingbo Zhou, Tong Xu, Dejing Dou, and Hui Xiong. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4541–4549, 2022.

Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pp. 417–428, 2023.

Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. *Advances in Neural Information Processing Systems*, 36:28748–28760, 2023.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *International conference on machine learning*, pp. 7192–7203. PMLR, 2021.

Xuan Son Nguyen. The gyro-structure of some matrix manifolds. *Advances in Neural Information Processing Systems*, 35:26618–26630, 2022.

Xuan Son Nguyen and Shuo Yang. Building neural networks on matrix manifolds: A gyrovector space approach. In *International Conference on Machine Learning*, pp. 26031–26062. PMLR, 2023.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407*, 2018.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

Jianzhong Qi, Zhuowei Zhao, Egemen Tanin, Tingru Cui, Neema Nassir, and Majid Sarvi. A graph and attentive multi-path convolutional network for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6548–6560, 2022.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pp. 412–422. Springer, 2018.

Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. *arXiv preprint arXiv:1911.08411*, 2019.

Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1717–1727, 2022.

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2120–2131, 2023.

Abraham A Ungar. *Beyond the Einstein addition law and its gyroscopic Thomas precession: The theory of gyrogroups and gyrovector spaces*. Springer, 2002.

Abraham Albert Ungar. *Analytic hyperbolic geometry: Mathematical foundations and applications*. World Scientific, 2005.

Abraham Albert Ungar. *Analytic hyperbolic geometry in n dimensions: An introduction*. CRC Press, 2014.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19206–19214, 2024.

Xuemei Wei, Yezheng Liu, Jianshan Sun, Yuanchun Jiang, Qifeng Tang, and Kun Yuan. Dual subgraph-based graph neural network for friendship prediction in location-based social networks. *ACM Transactions on Knowledge Discovery from Data*, 17(3):1–28, 2023.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022*, pp. 1070–1079, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pp. 5708–5717. PMLR, 2018.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. Graph-revised convolutional network. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 378–393. Springer, 2020.

Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. Generalized graph prompt: Toward a unification of pre-training and downstream tasks on graphs. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6237–6250, 2024.

Xingtong Yu, Jie Zhang, Yuan Fang, and Renhe Jiang. Non-homophilic graph pre-training and prompt learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1844–1854, 2025.

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 5829–5836, 2019.

Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2020.

Zhilun Zhou, Yu Liu, Jingtao Ding, Depeng Jin, and Yong Li. Hierarchical knowledge graph learning enabled socioeconomic indicator prediction in location-based social network. In *Proceedings of the ACM web conference 2023*, pp. 122–132, 2023.

APPENDIX

A Default Notation **15**

B Limitations **15**

C AI use statement **15**

D Preliminaries **15**

D.1 Cholesky Manifold 15

D.2 Gyrovector Spaces 16

E Riemannian Structure of Graph Manifold **17**

E.1 Proof of Pro.1 17

E.2 Proof of Pro.2 18

E.3 Proof of Pro.3 19

E.4 Proof of Pro.4 21

E.5 Proof of Pro.5 22

F Gyrovector Extension of Graph Manifold **23**

F.1 Proof of Lem. 1 23

F.2 Proof of Lem. 2 24

F.3 Proof of Lem. 3 24

F.4 Proof of The. 1 25

G Experimental Details **29**

G.1 Datasets 29

G.2 Pre-training Strategies 29

G.3 Implementation Details 30

A Default Notation

Table 3: Summary of notation.

Notation	Explanation
$\mathcal{A}^n$	Graph Manifold
$\mathcal{L}_+^n$	Cholesky manifold
$T_L\mathcal{L}_+^n$	Tangent space of the Cholesky manifold at L
$T_{A(L)}\mathcal{A}^n$	Tangent space of the Graph Manifold at $\mathcal{A}^n$
$T_{A(L)}^{\text{off}}\mathcal{A}^n$	Off-diagonal tangent subspace of the Graph Manifold at $\mathcal{A}^n$
$T_{A(L)}^{\text{diag}}\mathcal{A}^n$	Diagonal tangent subspace of the Graph Manifold at $\mathcal{A}^n$
$G = (V, E, A)$	A graph with node set V, edge set E, and adjacency matrix A
A	An adjacency matrix
V	Set of nodes in a graph
E	Set of edges in a graph
$[L]$	Strictly lower-triangular part of L
$D(L)$	Diagonal matrix containing the diagonal entries of L
$\text{diag}(L)$	Diagonal matrix containing the diagonal entries of L
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context

B Limitations

Although this work provides a unified continuous geometric representation for graph structures and further models their intrinsic distributions, it still has certain limitations. First, the current Graph Manifold is primarily constructed for undirected graphs and does not account for directional information in directed graphs. Since edges in directed graphs have explicit orientations and their topological structures differ fundamentally from those of undirected graphs, effectively encoding and characterising edge directions on the Graph Manifold remains an important issue when extending Graph Manifold to more general graph structures.

C AI use statement

Generative AI tools were used during the preparation of this work solely to assist with writing and language polishing, including improving language, grammar, and readability. The authors take full responsibility for the final content of this work.

D Preliminaries

D.1 Cholesky Manifold

Let $\mathcal{L}_+$ denote the Cholesky manifold, consisting of lower-triangular matrices with strictly positive diagonal entries:

$$\mathcal{L}_+ = \left\{ L \in \mathbb{R}^{m\times m} \mid L_{ij} = 0 \ (i < j),\ L_{ii} > 0 \right\}. \tag{22}$$

The tangent space at $L \in \mathcal{L}_+$ is given by

$$T_L\mathcal{L}_+ = \mathcal{L}, \tag{23}$$

where $\mathcal{L}$ denotes the space of lower-triangular matrices.

For tangent vectors $X, Y \in T_L\mathcal{L}_+$, the Log-Cholesky Riemannian metric is defined as

$$\widetilde{g}_L(X, Y) = \sum_{i>j} X_{ij}Y_{ij} + \sum_{j=1}^{m} X_{jj}Y_{jj}L_{jj}^{-2}. \tag{24}$$

Given $L \in \mathcal{L}_+$ and a tangent direction $X \in T_L\mathcal{L}_+$, the geodesic emanating from L in the direction X is

$$\widetilde{\gamma}_{L,X}(t) = [L] + t[X] + \mathrm{D}(L) \exp\left\{t\,\mathrm{D}(X)\,\mathrm{D}(L)^{-1}\right\}, \tag{25}$$

where $[L]$ and $\mathrm{D}(L)$ denote the strictly lower-triangular and diagonal components of L, respectively.

Accordingly, the Riemannian exponential map at L is

$$\widetilde{\mathrm{Exp}}_L(X) = [L] + [X] + \mathrm{D}(L) \exp\left\{\mathrm{D}(X)\,\mathrm{D}(L)^{-1}\right\}. \tag{26}$$

Conversely, the Riemannian logarithmic map from $K \in \mathcal{L}_+$ to the tangent space at L is given by

$$\widetilde{\mathrm{Log}}_L(K) = [K] - [L] + \mathrm{D}(L) \log\left\{\mathrm{D}(L)^{-1}\,\mathrm{D}(K)\right\}. \tag{27}$$

Finally, the corresponding geodesic distance between two points $L, K \in \mathcal{L}_+$ is

$$d_{\mathcal{L}_+}(L, K) = \left\{\|[L] - [K]\|_F^2 + \|\log \mathrm{D}(L) - \log \mathrm{D}(K)\|_F^2\right\}^{1/2}. \tag{28}$$

D.2 Gyrovector Spaces

This appendix provides the formal definitions and algebraic axioms underlying the gyrovector structure employed on the proposed Graph Manifold. We first introduce gyrogroups, followed by gyrocommutative gyrogroups and gyrovector spaces.

Definition 2 (Gyrogroups). *Let G be a non-empty set equipped with a binary operation $\oplus : G \times G \to G$. The pair $(G, \oplus)$ is called a* gyrogroup *if, for arbitrary $a, b, c \in G$, the following conditions are satisfied:*

(G1) There exists an element $e \in G$ such that

$$e \oplus a = a. \tag{29}$$

The element e is referred to as the left identity.

(G2) For every $a \in G$, there exists an element $\ominus a \in G$ such that

$$(\ominus a) \oplus a = e. \tag{30}$$

The element $\ominus a$ is called the left inverse of a.

(G3) For every $a, b \in G$, there exists an automorphism

$$\mathrm{gyr}[a, b] : G \to G \tag{31}$$

such that

$$a \oplus (b \oplus c) = (a \oplus b) \oplus \mathrm{gyr}[a, b]c. \tag{32}$$

This property is referred to as the left gyroassociative law*, and $\mathrm{gyr}[a, b]$ is called the* gyroautomorphism *or* gyration *generated by a and b.*

(G4) The gyroautomorphism satisfies the left reduction property

$$\mathrm{gyr}[a, b] = \mathrm{gyr}[a \oplus b, b]. \tag{33}$$

Definition 3 (Gyrocommutative Gyrogroups). *A gyrogroup $(G, \oplus)$ is called* gyrocommutative *if its binary operation satisfies*

$$a \oplus b = \mathrm{gyr}[a, b](b \oplus a), \qquad \forall a, b \in G. \tag{34}$$

Thus, unlike conventional commutativity, the operands cannot in general be interchanged directly. Instead, the discrepancy between the two orders is characterised by the corresponding gyroautomorphism.

Definition 4 (Gyrovector Spaces). *Let $(G, \oplus)$ be a gyrocommutative gyrogroup. Suppose that G is additionally equipped with a scalar multiplication*

$$\otimes : \mathbb{R} \times G \to G, \qquad (t, a) \mapsto t \otimes a. \tag{35}$$

The resulting structure $(G, \oplus, \otimes)$ is called a gyrovector space *if, for any $s, t \in \mathbb{R}$ and $a, b, c \in G$, the following conditions hold:*

(V1)

$$1 \otimes a = a, \qquad 0 \otimes a = e, \qquad (-1) \otimes a = \ominus a. \tag{36}$$

(V2)

$$(s + t) \otimes a = (s \otimes a) \oplus (t \otimes a). \tag{37}$$

(V3)

$$(st) \otimes a = s \otimes (t \otimes a). \tag{38}$$

(V4)

$$\mathrm{gyr}[a, b](t \otimes c) = t \otimes \mathrm{gyr}[a, b]c. \tag{39}$$

(V5)

$$\mathrm{gyr}[s \otimes a, t \otimes a] = \mathrm{Id}, \tag{40}$$

where Id *denotes the identity mapping on* G.

The above definitions establish the algebraic criteria required for a gyrovector structure. In the proposed Graph Manifold, we instantiate these abstract operations with specific manifold-valued constructions and verify the gyrogroup, gyrocommutative, and scalar-compatibility properties accordingly (Kim, 2015; 2016; 2020a;b). These properties provide the theoretical foundation for the algebraic operations introduced in the following sections.

E Riemannian Structure of Graph Manifold

E.1 Proof of Pro.1

Proof. Since L belongs to the Cholesky manifold $\mathcal{L}_+^n$, its tangent space consists of local variation directions that preserve the lower-triangular structure. Thus, for any $X \in T_L\mathcal{L}_+^n$, we have $X_{ij} = 0$ for $i < j$.

Consider the structural mapping from the Cholesky manifold to the graph adjacency matrix space,

$$\Phi(L) = L + L^\top - \mathrm{diag}(L). \tag{41}$$

Let $\gamma(t)$ be a smooth curve on $\mathcal{L}_+^n$ passing through L, satisfying

$$\gamma(0) = L, \qquad \dot{\gamma}(0) = X. \tag{42}$$

Substituting this curve into the structural mapping gives

$$\Phi(\gamma(t)) = \gamma(t) + \gamma(t)^\top - \mathrm{diag}(\gamma(t)). \tag{43}$$

Taking the derivative with respect to t at $t = 0$ yields

$$\mathrm{d}\Phi_L[X] = \dot{\gamma}(0) + \dot{\gamma}(0)^\top - \mathrm{diag}(\dot{\gamma}(0)). \tag{44}$$

Since $\dot{\gamma}(0) = X$, it follows that

$$\mathrm{d}\Phi_L[X] = X + X^\top - \mathrm{diag}(X). \tag{45}$$

Denoting the infinitesimal graph-topological variation induced by this differential as ΔA_X, we obtain

$$\Delta A_X = X + X^\top - \mathrm{diag}(X). \tag{46}$$

Therefore, all graph-topological variation directions induced by tangent vectors X through the differential of Φ constitute the tangent space of the Graph Manifold at $A(L)$, namely

$$T_{A(L)}\mathcal{A}^n = \left\{ \Delta A_X \mid X \in T_L\mathcal{L}_+^n \right\}. \tag{47}$$

Furthermore, since both $X + X^\top$ and $\mathrm{diag}(X)$ are symmetric, we have

$$\Delta A_X^\top = \left(X + X^\top - \mathrm{diag}(X)\right)^\top = X^\top + X - \mathrm{diag}(X) = \Delta A_X. \tag{48}$$

Hence, every tangent vector on the Graph Manifold is necessarily symmetric.

Finally, since X is lower triangular, for $i \geq j$ we have

$$(\Delta A_X)_{ij} = X_{ij}, \tag{49}$$

whereas for $i < j$, $X_{ij} = 0$ and therefore

$$(\Delta A_X)_{ij} = X_{ji}. \tag{50}$$

Thus, the differential of the structural mapping transforms the lower-triangular local variation direction X on the Cholesky manifold into a symmetric local graph-topological variation direction ΔA_X, thereby establishing the geometric correspondence between the tangent space of the Cholesky manifold and that of the Graph Manifold. □

E.2 Proof of Pro.2

Proof. According to the definition of the tangent space of the Graph Manifold, for any $\Delta A_X, \Delta A_Y \in T_{A(L)}\mathcal{A}^n$, there exist $X, Y \in T_L\mathcal{L}_+^n$ such that

$$\Delta A_X = X + X^\top - \mathrm{diag}(X), \tag{51}$$

and

$$\Delta A_Y = Y + Y^\top - \mathrm{diag}(Y). \tag{52}$$

Since X and Y belong to the tangent space of the Cholesky manifold at L, both are lower-triangular matrices. Moreover, according to the structural mapping, ΔA_X and ΔA_Y are symmetric.

We first consider the Frobenius inner product between the two complete Graph Manifold tangent vectors:

$$\langle \Delta A_X, \Delta A_Y \rangle_F = \mathrm{tr}\left(\Delta A_X^\top \Delta A_Y\right). \tag{53}$$

Since ΔA_X and ΔA_Y are symmetric, this can be equivalently written as

$$\langle \Delta A_X, \Delta A_Y \rangle_F = \mathrm{tr}\left(\Delta A_X \Delta A_Y\right). \tag{54}$$

For any independent off-diagonal position $i > j$, the structural mapping gives

$$(\Delta A_X)_{ij} = X_{ij}, \qquad (\Delta A_X)_{ji} = X_{ij}, \tag{55}$$

and similarly,

$$(\Delta A_Y)_{ij} = Y_{ij}, \qquad (\Delta A_Y)_{ji} = Y_{ij}. \tag{56}$$

Hence, the same undirected edge contributes twice to the complete Frobenius inner product:

$$\begin{aligned}(\Delta A_X)_{ij}(\Delta A_Y)_{ij} + (\Delta A_X)_{ji}(\Delta A_Y)_{ji} \\ = X_{ij}Y_{ij} + X_{ij}Y_{ij} = 2X_{ij}Y_{ij}.\end{aligned} \tag{57}$$

The factor 2 does not correspond to two independent geometric degrees of freedom. Instead, it results from the symmetric representation of an undirected graph, for which

$$A_{ij} = A_{ji}. \tag{58}$$

Thus, A_{ij} and A_{ji} represent the same undirected edge. To avoid double counting, we define the off-diagonal component as

$$\langle \Delta A_X, \Delta A_Y \rangle_{\mathrm{off}}^{\mathcal{A}} = \frac{1}{2}\sum_{i \neq j} (\Delta A_X)_{ij}(\Delta A_Y)_{ij}. \tag{59}$$

Since both tangent vectors are symmetric, this is equivalent to

$$\langle \Delta A_X, \Delta A_Y \rangle_{\mathrm{off}}^{\mathcal{A}} = \sum_{i > j} (\Delta A_X)_{ij}(\Delta A_Y)_{ij}. \tag{60}$$

Therefore, the off-diagonal component naturally corresponds to the independent edge variables of the undirected graph and is equipped with the standard Euclidean inner product.

We next consider the diagonal component. According to the structural mapping

$$\Phi(L) = L + L^\top - \mathrm{diag}(L), \tag{61}$$

the diagonal entries are not duplicated, and hence

$$(\Delta A_X)_{jj} = X_{jj}, \qquad (\Delta A_Y)_{jj} = Y_{jj}. \tag{62}$$

However, unlike the off-diagonal edge variables, the diagonal entries of the Cholesky factor satisfy the strict positivity constraint

$$L_{jj} > 0. \tag{63}$$

Consequently, the diagonal variables do not form unconstrained Euclidean coordinates. To preserve the intrinsic geometry associated with the positive diagonal structure, we adopt the Log-Cholesky geometry and define

$$\langle \Delta A_X, \Delta A_Y \rangle^{\mathcal{A}}_{\text{diag}} = \sum_{j=1}^{n} \frac{(\Delta A_X)_{jj}(\Delta A_Y)_{jj}}{L_{jj}^2}. \tag{64}$$

The tangent space of the Graph Manifold can therefore be decomposed as

$$T_{A(L)}\mathcal{A}^n = T^{\text{off}}_{A(L)}\mathcal{A}^n \oplus T^{\text{diag}}_{A(L)}\mathcal{A}^n. \tag{65}$$

The off-diagonal component describes local variations of the independent edge structures and is equipped with Euclidean geometry, whereas the diagonal component describes local variations of the positive diagonal structure and is equipped with Log-Cholesky geometry.

Combining these two components yields

$$\begin{aligned} g^{\mathcal{A}}_{A(L)}(\Delta A_X, \Delta A_Y) = &\langle \Delta A_X, \Delta A_Y \rangle^{\mathcal{A}}_{\text{off}} \\ &+ \langle \Delta A_X, \Delta A_Y \rangle^{\mathcal{A}}_{\text{diag}}. \end{aligned} \tag{66}$$

Substituting the two components gives

$$\begin{aligned} g^{\mathcal{A}}_{A(L)}(\Delta A_X, \Delta A_Y) = &\sum_{i>j} (\Delta A_X)_{ij}(\Delta A_Y)_{ij} \\ &+ \sum_{j=1}^{n} \frac{(\Delta A_X)_{jj}(\Delta A_Y)_{jj}}{L_{jj}^2}. \end{aligned} \tag{67}$$

Hence, the resulting Riemannian metric acts directly on the tangent vectors of the Graph Manifold, while X and Y provide their underlying representations through the differential of the structural mapping. The metric therefore jointly captures the intrinsic geometric properties of the independent graph-topological variations and the positive diagonal structure inherited from the Cholesky representation. □

E.3 Proof of Pro.3

Proof. According to the definition of the tangent space of the Graph Manifold, for any $\Delta A_X \in T_{A(L)}\mathcal{A}^n$, there exists $X \in T_L\mathcal{L}^n_+$ such that

$$\Delta A_X = X + X^\top - \text{diag}(X). \tag{68}$$

Here, X represents the underlying variation direction of the Cholesky factor, whereas ΔA_X denotes the corresponding tangent direction on the Graph Manifold.

According to the Riemannian metric defined above, the tangent space of the Graph Manifold consists of two components: the off-diagonal component equipped with Euclidean geometry and the diagonal component equipped with Log-Cholesky geometry. Therefore, the corresponding geodesic can be constructed according to these two geometric components.

For the initial point L and the initial direction X, the geodesic on the Cholesky manifold is given by

$$\widetilde{\gamma}_{L,X}(t) = [L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right). \tag{69}$$

The strictly lower-triangular component evolves linearly as

$$[\widetilde{\gamma}_{L,X}(t)] = [L] + t[X], \tag{70}$$

which follows from the Euclidean geometry adopted for the independent off-diagonal graph structures. Hence, the independent edge structures evolve continuously along their corresponding local topological directions.

The diagonal component evolves according to the Log-Cholesky geometry:

$$D\left(\widetilde{\gamma}_{L,X}(t)\right) = D(L)\exp\left(tD(X)D(L)^{-1}\right). \tag{71}$$

In particular, the j-th diagonal element evolves as

$$\left[\widetilde{\gamma}_{L,X}(t)\right]_{jj} = L_{jj}\exp\left(t\frac{X_{jj}}{L_{jj}}\right). \tag{72}$$

Since $L_{jj} > 0$ and the exponential function is strictly positive, we have

$$\left[\widetilde{\gamma}_{L,X}(t)\right]_{jj} > 0, \tag{73}$$

for all t. Therefore, the entire geodesic remains within the Cholesky manifold $\mathcal{L}_+^n$ and preserves the positive-diagonal constraint.

We then map the geodesic on the Cholesky manifold to the Graph Manifold through the structural mapping

$$\Phi(L) = L + L^\top - \mathrm{diag}(L). \tag{74}$$

Thus, the geodesic on the Graph Manifold is defined as

$$\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t) = \Phi\left(\widetilde{\gamma}_{L,X}(t)\right). \tag{75}$$

Applying the structural mapping explicitly gives

$$\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t) = \widetilde{\gamma}_{L,X}(t) + \widetilde{\gamma}_{L,X}(t)^\top - \mathrm{diag}\left(\widetilde{\gamma}_{L,X}(t)\right). \tag{76}$$

Substituting $\widetilde{\gamma}_{L,X}(t)$ yields

$$\begin{aligned}\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t) = &\left([L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right)\right)\\ &+ \left([L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right)\right)^\top\\ &- \mathrm{diag}\left([L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right)\right).\end{aligned} \tag{77}$$

Since the structural mapping produces a symmetric matrix for every t, and the diagonal entries of $\widetilde{\gamma}_{L,X}(t)$ remain strictly positive, the resulting curve remains within the Graph Manifold throughout its evolution.

At $t = 0$, we have

$$\widetilde{\gamma}_{L,X}(0) = [L] + D(L) = L. \tag{78}$$

Therefore,

$$\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(0) = \Phi(L) = A(L). \tag{79}$$

Moreover, differentiating the geodesic with respect to t gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\gamma}_{L,X}(t) + \left(\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\gamma}_{L,X}(t)\right)^\top - \mathrm{diag}\left(\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\gamma}_{L,X}(t)\right). \tag{80}$$

Since

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\gamma}_{L,X}(t)\bigg|_{t=0} = X, \tag{81}$$

we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t)\bigg|_{t=0} = X + X^\top - \mathrm{diag}(X) = \Delta A_X. \tag{82}$$

Hence, the geodesic satisfies the initial conditions

$$\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(0) = A(L), \qquad \dot{\gamma}^{\mathcal{A}}_{A(L),\Delta A_X}(0) = \Delta A_X. \tag{83}$$

Therefore, the geodesic describes the continuous evolution of the graph topology from the initial structure $A(L)$ along the prescribed local topological variation direction ΔA_X, namely,

$$A(L) \longrightarrow \gamma^{\mathcal{A}}_{A(L),\Delta A_X}(t). \tag{84}$$

The off-diagonal component corresponds to continuous variations of the independent edge structures and evolves linearly under Euclidean geometry, whereas the diagonal component corresponds to the positive diagonal structure of the Cholesky factor and evolves exponentially under Log-Cholesky geometry. Hence, the resulting geodesic is consistent with the tangent-space construction and the Riemannian metric defined on the Graph Manifold. □

E.4 Proof of Pro.4

Proof. According to the definition of the tangent space of the Graph Manifold, for any $\Delta A_X \in T_{A(L)}\mathcal{A}^n$, there exists $X \in T_L\mathcal{L}^n_+$ such that

$$\Delta A_X = X + X^\top - \mathrm{diag}(X). \tag{85}$$

Here, X serves as the underlying Cholesky representation of the tangent vector ΔA_X, whereas ΔA_X itself is the input to the Riemannian exponential map on the Graph Manifold.

According to the geodesic defined above, the Riemannian exponential map is obtained by evaluating the geodesic at unit time:

$$\mathrm{Exp}^{\mathcal{A}}_{A(L)}(\Delta A_X) = \gamma^{\mathcal{A}}_{A(L),\Delta A_X}(1). \tag{86}$$

Since the geodesic on the Graph Manifold is constructed through the structural mapping Φ, we have

$$\gamma^{\mathcal{A}}_{A(L),\Delta A_X}(1) = \Phi\left(\widetilde{\gamma}_{L,X}(1)\right). \tag{87}$$

The corresponding geodesic on the Cholesky manifold is

$$\widetilde{\gamma}_{L,X}(t) = [L] + t[X] + D(L)\exp\left(tD(X)D(L)^{-1}\right). \tag{88}$$

Evaluating it at $t = 1$ gives

$$\widetilde{\gamma}_{L,X}(1) = [L] + [X] + D(L)\exp\left(D(X)D(L)^{-1}\right). \tag{89}$$

We therefore define the underlying updated Cholesky representation as

$$L_{\mathrm{new}} = \widetilde{\gamma}_{L,X}(1). \tag{90}$$

Since $L \in \mathcal{L}^n_+$ and the diagonal component evolves as

$$[L_{\mathrm{new}}]_{jj} = L_{jj}\exp\left(\frac{X_{jj}}{L_{jj}}\right), \tag{91}$$

we have

$$[L_{\mathrm{new}}]_{jj} > 0, \tag{92}$$

for all j. Thus, L_{new} remains in the Cholesky manifold $\mathcal{L}^n_+$.

Applying the structural mapping to L_{new} yields the updated graph structure

$$A_{\mathrm{new}} = \Phi(L_{\mathrm{new}}) = L_{\mathrm{new}} + L^\top_{\mathrm{new}} - \mathrm{diag}(L_{\mathrm{new}}). \tag{93}$$

Hence,

$$\mathrm{Exp}^{\mathcal{A}}_{A(L)}(\Delta A_X) = A_{\mathrm{new}}. \tag{94}$$

Substituting the explicit expression of L_{new} gives

$$\begin{aligned} A_{\mathrm{new}} = &\Big([L] + [X] + D(L)\exp\left(D(X)D(L)^{-1}\right)\Big) \\ &+ \Big([L] + [X] + D(L)\exp\left(D(X)D(L)^{-1}\right)\Big)^\top \\ &- \mathrm{diag}\Big([L] + [X] + D(L)\exp\left(D(X)D(L)^{-1}\right)\Big). \end{aligned} \tag{95}$$

The structural mapping ensures that A_{new} is symmetric, while the positivity of the diagonal entries inherited from L_{new} ensures that the resulting point remains within the Graph Manifold. The exponential map itself is an operation defined directly from the tangent space of the Graph Manifold to the Graph Manifold, namely,

$$\mathrm{Exp}^{\mathcal{A}}_{A(L)} : T_{A(L)}\mathcal{A}^n \to \mathcal{A}^n, \tag{96}$$

where the input ΔA_X specifies the local graph-topological variation and the output A_{new} represents the resulting graph structure after finite Riemannian evolution. □

E.5 PROOF OF PRO.5

Proof. According to the definition of the tangent space of the Graph Manifold, for any $X \in T_L\mathcal{L}_+^n$, the corresponding tangent vector on the Graph Manifold is obtained through the differential of the structural mapping Φ as

$$d\Phi_L[X] = X + X^\top - \mathrm{diag}(X). \tag{97}$$

Therefore, to construct the tangent vector connecting $A(L)$ and $A(K)$, we first determine the corresponding Riemannian logarithmic map from L to K on the Cholesky manifold.

Under the Log-Cholesky geometry, the Riemannian logarithmic map from L to K is given by

$$\widetilde{\mathrm{Log}}_L(K) = [K] - [L] + D(L)\log\left(D(L)^{-1}D(K)\right). \tag{98}$$

We therefore denote the resulting Cholesky tangent vector by

$$X_{L\to K} = \widetilde{\mathrm{Log}}_L(K). \tag{99}$$

Here, $[K]-[L]$ describes the Euclidean variation of the strictly lower-triangular component, whereas

$$D(L)\log\left(D(L)^{-1}D(K)\right) \tag{100}$$

describes the intrinsic variation of the positive diagonal component under the Log-Cholesky geometry. Since $L, K \in \mathcal{L}_+^n$, their diagonal entries are strictly positive, and hence the above logarithmic operation is well defined.

Applying the differential of the structural mapping to $X_{L\to K}$ gives the corresponding tangent vector on the Graph Manifold,

$$\Delta A_{L\to K} = d\Phi_L\left[X_{L\to K}\right] = X_{L\to K} + X_{L\to K}^\top - \mathrm{diag}(X_{L\to K}). \tag{101}$$

Substituting the explicit expression of $X_{L\to K}$ yields

$$\begin{aligned}\Delta A_{L\to K} = &\Big([K] - [L] + D(L)\log\left(D(L)^{-1}D(K)\right)\Big)\\ &+ \Big([K] - [L] + D(L)\log\left(D(L)^{-1}D(K)\right)\Big)^\top\\ &- \mathrm{diag}\Big([K] - [L] + D(L)\log\left(D(L)^{-1}D(K)\right)\Big).\end{aligned} \tag{102}$$

By construction,

$$\Delta A_{L\to K} = \Delta A_{L\to K}^\top, \tag{103}$$

and hence

$$\Delta A_{L\to K} \in T_{A(L)}\mathcal{A}^n. \tag{104}$$

Thus, the Riemannian logarithmic map on the Graph Manifold is given by

$$\mathrm{Log}^{\mathcal{A}}_{A(L)}(A(K)) = \Delta A_{L\to K}. \tag{105}$$

Finally, according to the definition of the Riemannian exponential and logarithmic maps, the obtained tangent vector satisfies

$$\mathrm{Exp}^{\mathcal{A}}_{A(L)}\Big(\mathrm{Log}^{\mathcal{A}}_{A(L)}(A(K))\Big) = A(K), \tag{106}$$

provided that $A(K)$ lies within the corresponding normal neighbourhood of $A(L)$. Therefore, $\Delta A_{L\to K}$ represents the intrinsic local topological variation required to move from $A(L)$ to $A(K)$ along the Riemannian geometry of the Graph Manifold. This construction is consistent with the previously defined tangent space, geodesic, and Riemannian exponential map. □

F GYROVECTOR EXTENSION OF GRAPH MANIFOLD

F.1 PROOF OF LEM. 1

Proof. We derive the closed-form expression by successively evaluating the logarithmic map, parallel transport, and exponential map on the Graph Manifold.

First, according to the logarithmic map defined on $\mathcal{A}^n$, the tangent vector from the reference graph structure $E = A(L_E)$ to $A(K)$ is given by

$$\mathrm{Log}_E^{\mathcal{A}}(A(K)) = \Delta A_{E\to K}, \tag{107}$$

whose corresponding Cholesky tangent representation is

$$X_{E\to K} = [K] - [L_E] + D(L_E)\log\left(D(L_E)^{-1}D(K)\right). \tag{108}$$

Next, parallel transport this tangent vector from E to $A(L)$. Under the adopted metric, the transported tangent representation is

$$X_{E\to L;K} = [K] - [L_E] + D(L)\log\left(D(L_E)^{-1}D(K)\right). \tag{109}$$

Therefore, the corresponding tangent vector on the Graph Manifold is

$$\Delta A_{E\to L;K} = X_{E\to L;K} + X_{E\to L;K}^{\top} - \mathrm{diag}\left(X_{E\to L;K}\right). \tag{110}$$

We then apply the Riemannian exponential map at $A(L)$. In the corresponding Cholesky representation, the resulting point is

$$L_{\oplus} = [L] + [X_{E\to L;K}] + D(L)\exp\left(D(X_{E\to L;K})D(L)^{-1}\right). \tag{111}$$

Substituting the expression of $X_{E\to L;K}$ into the above equation gives

$$\begin{aligned} L_{\oplus} = {} & [L] + [K] - [L_E] \\ & + D(L)\log\left(D(L_E)^{-1}D(K)\right) \\ & + D(L)\exp\left(D(X_{E\to L;K})D(L)^{-1}\right). \end{aligned} \tag{112}$$

Since the diagonal component of $X_{E\to L;K}$ is

$$D(X_{E\to L;K}) = D(K) - D(L_E) + D(L)\log\left(D(L_E)^{-1}D(K)\right), \tag{113}$$

the diagonal contribution in the exponential map simplifies to

$$D(L)\exp\left(D(X_{E\to L;K})D(L)^{-1}\right) = D(L)D(L_E)^{-1}D(K). \tag{114}$$

Consequently, the Cholesky representation of the gyro sum is

$$L_{\oplus} = [L] + [K] - [L_E] + D(L)D(L_E)^{-1}D(K). \tag{115}$$

Finally, applying the structural mapping

$$A(L_{\oplus}) = L_{\oplus} + L_{\oplus}^{\top} - \mathrm{diag}(L_{\oplus}), \tag{116}$$

we obtain

$$\begin{aligned} A(L)\oplus_{\mathcal{A}} A(K) = {} & [L] + [K] - [L_E] \\ & + \left([L] + [K] - [L_E]\right)^{\top} \\ & + D(L)D(L_E)^{-1}D(K). \end{aligned} \tag{117}$$

This completes the proof. □

F.2 Proof of Lem. 2

Proof. By definition, the gyro inverse of $A(L)$ is the graph structure $\ominus_{\mathcal{A}} A(L)$ satisfying

$$A(L) \oplus_{\mathcal{A}} (\ominus_{\mathcal{A}} A(L)) = E. \tag{118}$$

Let

$$\ominus_{\mathcal{A}} A(L) = A(\widetilde{L}), \tag{119}$$

where $\widetilde{L}$ denotes the Cholesky representation of the unknown inverse. Using the closed-form expression of the gyro addition, we have

$$A(L) \oplus_{\mathcal{A}} A(\widetilde{L}) = A(L_{\oplus}), \tag{120}$$

with

$$L_{\oplus} = [L] + [\widetilde{L}] - [L_E] + D(L)D(L_E)^{-1}D(\widetilde{L}). \tag{121}$$

Since the gyro inverse is required to satisfy

$$A(L) \oplus_{\mathcal{A}} A(\widetilde{L}) = A(L_E), \tag{122}$$

it is sufficient to impose

$$L_{\oplus} = L_E. \tag{123}$$

For the strict lower-triangular component, this gives

$$[L] + [\widetilde{L}] - [L_E] = [L_E], \tag{124}$$

and hence

$$[\widetilde{L}] = 2[L_E] - [L]. \tag{125}$$

For the diagonal component, we require

$$D(L)D(L_E)^{-1}D(\widetilde{L}) = D(L_E). \tag{126}$$

Multiplying the above equation from the left by $D(L)^{-1}$ and from the right by $D(L_E)^{-1}$ yields

$$D(\widetilde{L}) = D(L_E)D(L)^{-1}D(L_E). \tag{127}$$

Combining the strict lower-triangular and diagonal components gives

$$\widetilde{L} = 2[L_E] - [L] + D(L_E)D(L)^{-1}D(L_E). \tag{128}$$

Therefore,

$$\ominus_{\mathcal{A}} A(L) = A\left(2[L_E] - [L] + D(L_E)D(L)^{-1}D(L_E)\right). \tag{129}$$

Thus, the gyro inverse is given by

$$L_{\mathcal{A}}^{-1} = 2[L_E] - [L] + D(L_E)D(L)^{-1}D(L_E). \tag{130}$$

□

F.3 Proof of Lem. 3

Proof. By definition, the gyro scalar multiplication is obtained by first mapping $A(L)$ to the tangent space at the reference graph E, scaling the resulting tangent vector by t, and then mapping it back to the Graph Manifold through the exponential map.

According to the logarithmic map defined on $\mathcal{A}^n$, the Cholesky tangent representation of $\mathrm{Log}_E^{\mathcal{A}}(A(L))$ is

$$X_{E\to L} = [L] - [L_E] + D(L_E)\log\left(D(L_E)^{-1}D(L)\right). \tag{131}$$

Scaling this tangent representation by t gives

$$\begin{aligned} tX_{E\to L} = {} & t([L] - [L_E]) \\ & + tD(L_E)\log\left(D(L_E)^{-1}D(L)\right). \end{aligned} \tag{132}$$

We next apply the Riemannian exponential map at the reference graph $E = A(L_E)$. The corresponding Cholesky representation is

$$\begin{aligned} L_t = [L_E] + t([L] - [L_E]) \\ + D(L_E) \exp\left(t \log\left(D(L_E)^{-1} D(L)\right)\right). \end{aligned} \tag{133}$$

Using the identity

$$\exp(t \log X) = X^t, \tag{134}$$

for a positive diagonal matrix X, we obtain

$$D(L_E) \exp\left(t \log\left(D(L_E)^{-1} D(L)\right)\right) = D(L_E) \left(D(L_E)^{-1} D(L)\right)^t. \tag{135}$$

Consequently,

$$L_t = [L_E] + t([L] - [L_E]) + D(L_E) \left(D(L_E)^{-1} D(L)\right)^t. \tag{136}$$

Therefore,

$$t \odot_{\mathcal{A}} A(L) = A(L_t). \tag{137}$$

Finally, applying the structural mapping

$$A(L_t) = L_t + L_t^\top - \mathrm{diag}(L_t), \tag{138}$$

and separating the strict lower-triangular and diagonal components of L_t, we obtain

$$\begin{aligned} t \odot_{\mathcal{A}} A(L) = [L_E] + t([L] - [L_E]) \\ + ([L_E] + t([L] - [L_E]))^\top \\ + D(L_E) \left(D(L_E)^{-1} D(L)\right)^t. \end{aligned} \tag{139}$$

This completes the proof. □

F.4 Proof of The. 1

Proof. For any $A(L), A(K), A(R) \in \mathcal{A}^n$, recall that the gyro addition on the Graph Manifold is given by

$$A(L) \oplus_{\mathcal{A}} A(K) = A(L_\oplus), \tag{140}$$

where

$$L_\oplus = [L] + [K] - [L_E] + D(L) D(L_E)^{-1} D(K), \tag{141}$$

and $E = A(L_E)$ denotes the identity element.

We also recall that the gyro inverse is given by

$$\ominus_{\mathcal{A}} A(L) = A\left(L_{\mathcal{A}}^{-1}\right), \tag{142}$$

where

$$L_{\mathcal{A}}^{-1} = 2[L_E] - [L] + D(L_E) D(L)^{-1} D(L_E), \tag{143}$$

and the gyro scalar multiplication is given by

$$t \odot_{\mathcal{A}} A(L) = A(L_t), \tag{144}$$

where

$$L_t = [L_E] + t([L] - [L_E]) + D(L_E) \left(D(L_E)^{-1} D(L)\right)^t. \tag{145}$$

For convenience, define the intrinsic coordinate representation

$$\Xi(L) = \left([L] - [L_E],\ \log D(L) - \log D(L_E)\right). \tag{146}$$

Since the strict lower-triangular component and the diagonal component uniquely determine L, the mapping Ξ is injective. Moreover, from the explicit expressions above, we have

$$\Xi(L_\oplus) = \Xi(L) + \Xi(K), \tag{147}$$

and

$$\Xi(L_t) = t\,\Xi(L). \tag{148}$$

These two relations will be used repeatedly below.

(G1) Left Identity.

We first verify that $E = A(L_E)$ is a left identity. From the explicit expression of the gyro addition,

$$E \oplus_{\mathcal{A}} A(L) = A(L_{E\oplus L}), \tag{149}$$

where

$$\begin{aligned} L_{E\oplus L} &= [L_E] + [L] - [L_E] + D(L_E)D(L_E)^{-1}D(L) \\ &= [L] + D(L) \\ &= L. \end{aligned} \tag{150}$$

Therefore,

$$E \oplus_{\mathcal{A}} A(L) = A(L), \tag{151}$$

and hence E is a left identity of $(\mathcal{A}^n, \oplus_{\mathcal{A}})$.

(G2) Left Inverse.

We next verify that the explicitly defined gyro inverse is a left inverse. Using

$$\ominus_{\mathcal{A}} A(L) = A\left(L_{\mathcal{A}}^{-1}\right), \tag{152}$$

we obtain

$$\begin{aligned} &\ominus_{\mathcal{A}} A(L) \oplus_{\mathcal{A}} A(L) \\ &= A\Big([L_{\mathcal{A}}^{-1}] + [L] - [L_E] + D(L_{\mathcal{A}}^{-1})D(L_E)^{-1}D(L)\Big). \end{aligned} \tag{153}$$

For the strict lower-triangular component,

$$\begin{aligned} [L_{\mathcal{A}}^{-1}] + [L] - [L_E] &= 2[L_E] - [L] + [L] - [L_E] \\ &= [L_E]. \end{aligned} \tag{154}$$

For the diagonal component,

$$\begin{aligned} &D(L_{\mathcal{A}}^{-1})D(L_E)^{-1}D(L) \\ &= D(L_E)D(L)^{-1}D(L_E)D(L_E)^{-1}D(L) \\ &= D(L_E). \end{aligned} \tag{155}$$

Consequently,

$$\ominus_{\mathcal{A}} A(L) \oplus_{\mathcal{A}} A(L) = A(L_E) = E. \tag{156}$$

Thus, $\ominus_{\mathcal{A}} A(L)$ is a left inverse of $A(L)$.

Left Cancellation Law.

The left cancellation law follows directly from the same explicit expressions. Let

$$A(L) \oplus_{\mathcal{A}} A(K) = A(L_{\oplus}). \tag{157}$$

Then

$$\begin{aligned} &\ominus_{\mathcal{A}} A(L) \oplus_{\mathcal{A}} (A(L) \oplus_{\mathcal{A}} A(K)) \\ &= A\Big([L_{\mathcal{A}}^{-1}] + [L_{\oplus}] - [L_E] + D(L_{\mathcal{A}}^{-1})D(L_E)^{-1}D(L_{\oplus})\Big). \end{aligned} \tag{158}$$

For the strict lower-triangular component,

$$\begin{aligned} [L_{\mathcal{A}}^{-1}] + [L_{\oplus}] - [L_E] &= (2[L_E] - [L]) \\ &\quad + ([L] + [K] - [L_E]) - [L_E] \\ &= [K]. \end{aligned} \tag{159}$$

For the diagonal component,

$$\begin{aligned} &D(L_{\mathcal{A}}^{-1})D(L_E)^{-1}D(L_{\oplus}) \\ &= D(L_E)D(L)^{-1}D(L_E)D(L_E)^{-1}D(L)D(L_E)^{-1}D(K) \\ &= D(K). \end{aligned} \tag{160}$$

Hence,

$$\ominus_{\mathcal{A}} A(L) \oplus_{\mathcal{A}} (A(L) \oplus_{\mathcal{A}} A(K)) = A(K). \tag{161}$$

Therefore, the Left Cancellation Law holds.

Associativity of the Gyro Addition.

Before verifying the remaining gyrogroup axioms, we establish the associativity of $\oplus_{\mathcal{A}}$. By the definition of Ξ,

$$\Xi(L_{\oplus}) = \Xi(L) + \Xi(K). \tag{162}$$

Therefore,

$$\begin{aligned} \Xi\left((L \oplus K) \oplus R\right) &= \Xi(L_{\oplus}) + \Xi(R) \\ &= \Xi(L) + \Xi(K) + \Xi(R), \end{aligned} \tag{163}$$

whereas

$$\begin{aligned} \Xi\left(L \oplus (K \oplus R)\right) &= \Xi(L) + \Xi(K_{\oplus}) \\ &= \Xi(L) + \Xi(K) + \Xi(R). \end{aligned} \tag{164}$$

Since Ξ is injective, it follows that

$$(A(L) \oplus_{\mathcal{A}} A(K)) \oplus_{\mathcal{A}} A(R) = A(L) \oplus_{\mathcal{A}} (A(K) \oplus_{\mathcal{A}} A(R)). \tag{165}$$

Thus, the gyro addition is associative.

(G3) Left Gyroassociative Law.

By definition,

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{Id}_{\mathcal{A}}. \tag{166}$$

Hence,

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]A(R) = A(R). \tag{167}$$

Using the associativity established above, we obtain

$$\begin{aligned} &A(L) \oplus_{\mathcal{A}} (A(K) \oplus_{\mathcal{A}} A(R)) \\ &= (A(L) \oplus_{\mathcal{A}} A(K)) \oplus_{\mathcal{A}} A(R) \\ &= (A(L) \oplus_{\mathcal{A}} A(K)) \oplus_{\mathcal{A}} \mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]A(R). \end{aligned} \tag{168}$$

Therefore, the Left Gyroassociative Law (G3) holds.

Moreover, since $\mathrm{Id}_{\mathcal{A}}$ is an automorphism of $\mathcal{A}^n$, the gyroautomorphism required in (G3) is indeed an automorphism.

(G4) Left Reduction Property.

For arbitrary $A(L), A(K) \in \mathcal{A}^n$, we have

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{Id}_{\mathcal{A}}. \tag{169}$$

Similarly,

$$\mathrm{gyr}_{\mathcal{A}}[A(L) \oplus_{\mathcal{A}} A(K), A(K)] = \mathrm{Id}_{\mathcal{A}}. \tag{170}$$

Consequently,

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{gyr}_{\mathcal{A}}[A(L) \oplus_{\mathcal{A}} A(K), A(K)]. \tag{171}$$

Thus, the Left Reduction Property (G4) is satisfied.

Therefore,

$$(\mathcal{A}^n, \oplus_{\mathcal{A}}) \tag{172}$$

forms a gyrogroup.

Gyrocommutative Law.

It remains to verify that the gyrogroup is gyrocommutative. From the explicit expression of the gyro addition,

$$L_{\oplus} = [L] + [K] - [L_E] + D(L)D(L_E)^{-1}D(K). \tag{173}$$

Since the strict lower-triangular components are added commutatively and all diagonal matrices commute, we have

$$D(L)D(L_E)^{-1}D(K) = D(K)D(L_E)^{-1}D(L). \tag{174}$$

Hence,

$$L_{\oplus}(L, K) = L_{\oplus}(K, L), \tag{175}$$

and therefore

$$A(L) \oplus_{\mathcal{A}} A(K) = A(K) \oplus_{\mathcal{A}} A(L). \tag{176}$$

Since

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{Id}_{\mathcal{A}}, \tag{177}$$

we obtain

$$\begin{aligned} A(L) \oplus_{\mathcal{A}} A(K) &= A(K) \oplus_{\mathcal{A}} A(L) \\ &= \mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]\,(A(K) \oplus_{\mathcal{A}} A(L))\,. \end{aligned} \tag{178}$$

Thus, the gyrocommutative law holds.

(V1) Identity, Zero Scalar, and Negative Unit Scalar.

We first consider the scalar multiplication at $t = 0$. From the explicit expression of L_t,

$$\begin{aligned} L_0 &= [L_E] + D(L_E)\left(D(L_E)^{-1}D(L)\right)^0 \\ &= [L_E] + D(L_E) \\ &= L_E. \end{aligned} \tag{179}$$

Therefore,

$$0 \odot_{\mathcal{A}} A(L) = A(L_E) = E. \tag{180}$$

For $t = 1$, we have

$$\begin{aligned} L_1 &= [L_E] + [L] - [L_E] + D(L_E)\left(D(L_E)^{-1}D(L)\right) \\ &= [L] + D(L) \\ &= L. \end{aligned} \tag{181}$$

Hence,

$$1 \odot_{\mathcal{A}} A(L) = A(L). \tag{182}$$

For $t = -1$, we obtain

$$\begin{aligned} L_{-1} &= [L_E] - ([L] - [L_E]) + D(L_E)\left(D(L_E)^{-1}D(L)\right)^{-1} \\ &= 2[L_E] - [L] + D(L_E)D(L)^{-1}D(L_E) \\ &= L_{\mathcal{A}}^{-1}. \end{aligned} \tag{183}$$

Consequently,

$$(-1) \odot_{\mathcal{A}} A(L) = A(L_{\mathcal{A}}^{-1}) = \ominus_{\mathcal{A}} A(L). \tag{184}$$

Thus, (V1) holds.

(V2) Compatibility with Scalar Addition.

Using the intrinsic coordinate representation,

$$\Xi(L_t) = t\Xi(L). \tag{185}$$

Therefore,

$$\begin{aligned} \Xi\left(L_{s+t}\right) &= (s+t)\Xi(L) \\ &= s\Xi(L) + t\Xi(L) \\ &= \Xi(L_s) + \Xi(L_t) \\ &= \Xi\left(L_s \oplus L_t\right). \end{aligned} \tag{186}$$

Since Ξ is injective,

$$L_{s+t} = L_s \oplus L_t. \tag{187}$$

Hence,

$$(s+t) \odot_{\mathcal{A}} A(L) = s \odot_{\mathcal{A}} A(L) \oplus_{\mathcal{A}} t \odot_{\mathcal{A}} A(L). \tag{188}$$

Thus, (V2) is satisfied.

(V3) Compatibility with Scalar Multiplication.

Similarly,

$$\begin{aligned} \Xi\left(L_{st}\right) &= st\Xi(L) \\ &= s\left(t\Xi(L)\right) \\ &= s\Xi(L_t) \\ &= \Xi\left(L_{t,s}\right), \end{aligned} \tag{189}$$

where $L_{t,s}$ denotes the Cholesky representation associated with $s \odot_{\mathcal{A}} (t \odot_{\mathcal{A}} A(L))$. By injectivity of Ξ,

$$L_{st} = L_{t,s}. \tag{190}$$

Consequently,

$$(st) \odot_{\mathcal{A}} A(L) = s \odot_{\mathcal{A}} \left(t \odot_{\mathcal{A}} A(L)\right). \tag{191}$$

Thus, (V3) holds.

(V4) Compatibility of Gyroautomorphisms with Scalar Multiplication.

Since the gyroautomorphism is the identity mapping,

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)] = \mathrm{Id}_{\mathcal{A}}, \tag{192}$$

we have

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]\left(t \odot_{\mathcal{A}} A(R)\right) = t \odot_{\mathcal{A}} A(R), \tag{193}$$

while

$$t \odot_{\mathcal{A}} \left(\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]A(R)\right) = t \odot_{\mathcal{A}} A(R). \tag{194}$$

Therefore,

$$\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]\left(t \odot_{\mathcal{A}} A(R)\right) = t \odot_{\mathcal{A}} \left(\mathrm{gyr}_{\mathcal{A}}[A(L), A(K)]A(R)\right). \tag{195}$$

Thus, (V4) is satisfied.

(V5) Trivial Gyroautomorphism Along Scalar Multiples.

For any $s, t \in \mathbb{R}$ and $A(L) \in \mathcal{A}^n$, the gyroautomorphism is the identity for every pair of elements in $\mathcal{A}^n$. Hence,

$$\mathrm{gyr}_{\mathcal{A}}\left[s \odot_{\mathcal{A}} A(L), t \odot_{\mathcal{A}} A(L)\right] = \mathrm{Id}_{\mathcal{A}}. \tag{196}$$

Therefore, (V5) holds.

Combining (G1)–(G4), the gyrocommutative law, and (V1)–(V5), we conclude that

$$(\mathcal{A}^n, \oplus_{\mathcal{A}}, \odot_{\mathcal{A}}) \tag{197}$$

forms a gyrovector space. Hence, the Graph Manifold equipped with the proposed gyro addition and gyro scalar multiplication constitutes a Graph Gyrovector Space. □

G Experimental Details

G.1 Datasets

G.2 Pre-training Strategies

We consider four commonly used pre-training paradigms for GNNs in our experiments, following the settings adopted in recent graph prompting studies (Liu et al., 2023; Sun et al., 2022; 2023). These paradigms cover both contrastive learning and link-prediction-based objectives, providing a diverse set of pre-trained GNNs for evaluating the effectiveness of our method.

Table 4: Basic information and statistics of graph datasets adopted in our experiments.

Dataset	#(Nodes)	#(Edges)	#(Features)	Average degree	Homophily ratio	#(Classes)
Cora	2,708	10,556	1,433	3.90	0.810	7
PubMed	19,717	88,648	500	4.49	0.802	3
Amazon	24,492	93,050	300	3.80	0.380	5
Minesweeper	10,000	39,402	7	3.94	0.683	2

- **GraphCL.** GraphCL obtains multiple graph views by applying structural augmentations, such as node dropping and edge perturbation. Each view is processed by the same GNN encoder, followed by a nonlinear projection head that maps the resulting representations into a contrastive embedding space. The pre-training objective encourages representations from the two augmented views of the same graph to remain close to each other, thereby learning augmentation-invariant graph representations.
- **SimGRACE.** Instead of modifying the input graph, SimGRACE introduces perturbations directly into the parameters of the GNN encoder. Specifically, Gaussian noise is added to the model parameters to construct a perturbed encoder. The original and perturbed encoders are then applied to the same graph, and their corresponding representations are treated as a positive pair in a contrastive learning objective.
- **LP-GPPT.** LP-GPPT adopts a link prediction objective for GNN pre-training. During training, a portion of the edges is masked from the input graph, and the model is trained to recover the connectivity between node pairs. Node pairs corresponding to existing edges serve as positive examples, whereas disconnected node pairs are used as negative examples. This objective encourages the pre-trained GNN to capture structural dependencies between nodes.
- **LP-GraphPrompt.** LP-GraphPrompt also formulates GNN pre-training as a link prediction problem, but constructs training pairs at the node level. For each target node, a connected neighbour is selected as a positive counterpart, while an unconnected node is sampled as a negative counterpart. The resulting objective promotes high similarity between structurally connected nodes and low similarity between disconnected nodes.

These four pre-training paradigms provide complementary mechanisms for learning graph representations. GraphCL and SimGRACE introduce perturbations to the graph views and model parameters, respectively, and learn representations through contrastive objectives. In contrast, LP-GPPT and LP-GraphPrompt explicitly exploit graph connectivity through link-prediction-based supervision.

It is worth noting that identifying the most effective pre-training strategy for GNNs is itself an important research problem. Existing graph prompting approaches employ different pre-training objectives, with link prediction and contrastive learning being two widely adopted choices (Liu et al., 2023; Sun et al., 2022; Yu et al., 2024).Recent studies have further investigated the conditions under which different pre-training paradigms are preferable (Yu et al., 2025). However, the selection or optimisation of the pre-training strategy is not the primary focus of this work. Instead, we evaluate our graph prompting approach under a range of representative pre-trained GNNs and concentrate on improving downstream performance given a fixed pre-trained model. In this way, the effectiveness of the proposed method can be assessed independently of any particular pre-training paradigm.

G.3 Implementation Details

For each graph prompting method, we employ a 2-layer GCN (Kipf & Welling, 2016a) as the GNN model, with a hidden dimension of 128. All experiments are conducted under the 5-shot setting with 500 training epochs. Each method is trained using the Riemannian Adam optimiser (Kingma & Ba, 2014) with a learning rate of 0.005. The reported performance is averaged over five runs with different random seeds.

During graph fine-tuning, we extract 2-hop subgraphs from the original graphs for local graph fine-tuning, and map the entire graph structure optimisation process onto the proposed Graph Manifold, such that graph topology updates are performed within its intrinsic geometry. Specifically, we model

graph topology variations probabilistically in the tangent space of the Graph Manifold and characterise the posterior distribution of graph structures using learnable mean and variance parameters. This enables the model to simultaneously capture the underlying directions of topological variation and the associated structural uncertainty.

During training, we draw multiple samples from the learned posterior distribution and map the sampled topology variations back to the Graph Manifold through the Riemannian exponential map, thereby obtaining multiple plausible graph structures for GNN prediction. During inference, we average the predictions obtained from different sampled graph structures and perform Monte Carlo marginalisation to aggregate information across multiple plausible topologies. This procedure explicitly accounts for graph structural uncertainty and leads to more robust predictions. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB of GPU memory.